LUERS

MICRO APPLICATION

2

AMSTRAD

PROGRAMMES BASIC POUR LES CPC



UN LIVRE DATA BECKER

LUERS

MICRO APPLICATION

2

AMSTRAD

PROGRAMMES BASIC POUR LES CPC



UN LIVRE DATA BECKER

Distribué par : MICRO APPLICATION
13, Rue Sainte Cécile
75009 PARIS

et

EDITION RADIO
9, Rue Jacob
75006 PARIS

ISBN : 2-86899-007-X

Traduction Française assurée par Pascal HAUSMAN

Collection dirigée par Mr Philippe OLIVIER
Edition réalisée par Frédérique BEAUDONNET

TABLE DES MATIERES

V. DES PROGRAMMES UTILES

VI. JEUX

VII. VOS PREMIERS PAS EN LANGAGE MACHINE

VIII. ANNEXE

Chère lectrice!

Cher lecteur!

Chère programmeuse

Cher programmeur!

Vous avez en main le livre 'Programmes BASIC' et vous voudriez peut-être pouvoir porter un premier jugement sur cet ouvrage. C'est pourquoi vous avez déjà jeté un coup d'oeil à la table des matières, après quoi vous vous préparez à sauter l'introduction et à vous diriger immédiatement vers la pièce tranquille où vous vous livrez à la programmation ou à l'écriture de programmes sur votre CPC 464.

J'aimerais toutefois pouvoir dire quelques mots avant que vous ne vous lanciez dans votre occupation favorite.

Le principe qui a en effet présidé à la réalisation de cette collection de programmes est différent de ce qu'on pourrait imaginer au premier abord. Cette collection de programmes ne comprend pas que des programmes mais également 40 pages de textes. La place occupée par ce texte n'a pas pour but de rémédier à un manque d'idées: si j'avais publié tous les programmes que j'ai réalisés, cet ouvrage comporterait autour de 400 pages!. Le texte est donc là pour que vous ne vous contentiez pas de taper des programmes à longueur de journée mais pour que vous puissiez aussi apprendre un certain nombre de choses sur le CPC 464 en tapant ces programmes.

Nous n'avons pas expliqué chaque ligne de programme pour ne pas mettre votre patience à trop rude épreuve et parce que nous n'en

avions pas la place, mais nous attirons votre attention dans les textes introductifs sur des détails intéressants et les programmes sont également richement pourvus en lignes REM explicatives.

En ce qui concerne le contenu, c'est surtout le côté pratique pour l'utilisateur qui a été favorisé, soit que vous vouliez éditer de la musique, du graphisme ou des textes, soit que vous ayez besoin de gérer votre stock de disques ou les résultats du championnat de football. Un autre point important de cet ouvrage est enfin de vous permettre d'avoir une idée de la façon dont travaille votre CPC. Le "désassembleur" ou la "liste des références de variables" ne constituent que deux exemples à cet égard.

En tout cas, et ce sera ma conclusion, nous souhaitons que vous aurez beaucoup de plaisir à pénétrer plus profondément dans l'univers des ordinateurs en général et du CPC 464 en particulier et que, peut-être, vous deviendrez grâce à cette collection de programmes un programmeur en herbe.

L'auteur des programmes et du texte vous souhaite encore une fois beaucoup de plaisir dans votre travail avec cet ouvrage.

Mémoire 1

=========

Si nous examinons à l'écran le résultat de ce programme, nous constatons que la zone de la mémoire dont nous demandons le contenu avec l'instruction "PEEK", n'est pas vide, puisqu'elle ne contient pas que des 0.

Ce que nous voyons à l'écran nous semble une suite de chiffres sans signification, mais nous allons pouvoir constater que derrière ces chiffres se cachent aussi des lettres et des caractères très précis. Faisons une petite expérience:

Entrez dans votre CPC: 'PRINT ASC("!")' (ENTER). Résultat: 33. Donc, derrière le signe d'exclamation se cache pour l'ordinateur le nombre 33. Faisons maintenant l'expérience dans l'autre sens; entrez: 'PRINT CHR$(33)' (ENTER).
Résultat: '!'. Nous avons donc demandé à l'ordinateur de convertir à nouveau le nombre 33 en un caractère. Le résultat est notre signe d'exclamation!

Nous pouvons maintenant apprendre à interpréter de même la masse de chiffres que nous avons obtenue avec le programme Mémoire 1. Pour le vérifier, veuillez entrer à la suite le programme 'Mémoire 2'.

Indication pour l'utilisation du programme 'Mémoire 1':

Eviter de choisir des adresses de début et de fin de la zone de la mémoire distantes de plus de 150 octets, de façon à ce que toutes les données que vous voulez examiner apparaissent sur une seule page d'écran.

```
10 REM Examen de la memoire 1
20 REM Programmes Basic sur CPC464
30 REM Copyright 1984 DATA BECKER &
                      Rainer Lueers
40 INK 0,1:INK 1,24:INK 2,1,24:effet=2:n
ormal=1
50 MODE 1
60 REM Delimiter cadre de l'examen
70 INPUT "Entrer adresse de debut ";a
80 INPUT "Entrer adresse de fin ";e
90 IF e<a OR e>65535 OR a<0 OR a>65535 T
HEN GOSUB 210:GOTO 50
100 PRINT
110 REM Sortie du contenu de la memoire
dans le cadre delimite
120 FOR n=a TO e
130 PRINT PEEK(n);
140 NEXT n
150 PRINT
160 PRINT
170 INPUT "Voulez-vous examiner d'autres
 parties dela memoire ( /N) ";f$
180 f$=UPPER$(f$)
190 IF f$<>"N" THEN GOTO 10
200 END
210 PEN effet:PRINT:PRINT TAB(12) "Entre
e incorrecte!"
220 PEN normal:GOSUB 230:RETURN
230 PRINT:PRINT TAB(7) "(Appuyez sur une
 touche S.V.P.)"
240 f$=INKEY$:IF f$="" THEN GOTO 240
250 RETURN
```

Mémoire 2

=========

Une fois que vous avez tapé ce programme et que vous l'avez lancé avec RUN, vous pouvez constater, suivant ce que vous faites, soit qu'il ne se passe rien, soit qu'il ne se passe que des évènements imprévus (l'écran se vide, le mode change, PEN et PAPER se modifient, etc...). Pourquoi?

Comme nous vous l'indiquons également dans d'autres programmes de ce recueil, il y a des caractères directement représentables (de CHR$(32)=espace à CHR$(126)=ligne zig-zag et même jusqu'à CHR$(255) si on compte les caractères graphiques) et des caractères qu'il n'est pas si facile de faire afficher (par exemple CHR$(2)=déconnexion du curseur texte ou CHR$(7)=faire entendre un bip, etc...).

Il nous faut donc écrire maintenant encore un troisième programme qui n'affichera à l'écran que les caractères ayant une fonction purement graphique. C'est ainsi que nous aboutissons en complétant un peu notre programme au programme Mémoire 3.

Indication pour l'utilisation du programme Mémoire 2:

Si vous ne voyez à l'écran plus aucune forme reconnaissable et que vous vouliez malgré tout continuer à utiliser le programme que vous venez d'entrer, il vous faut faire preuve d'un peu d'astuce: ouvrez le manuel de votre ordinateur au Chapitre 9 et essayez d'annuler les facteurs perturbateurs en entrant PRINT et le code CHR$ qui convient.

```
10 REM Examen de la memoire 2
20 REM Programmes Basic sur CPC464
30 REM Copyright 1984 DATA BECKER &
                       Rainer Lueers
40 INK 0,1:INK 1,24:INK 2,1,24:effet=2:n
ormal=1
50 MODE 1
60 REM Delimiter cadre de l'examen
70 INPUT "Entrer adresse de debut ";a
80 INPUT "Entrer adresse de fin ";e
90 IF e<a OR e>65535 OR a<0 OR a>65535 T
HEN GOSUB 210:GOTO 50
100 PRINT
110 REM Sortie du contenu de la memoire
dans le cadre delimite
120 FOR n=a TO e
130 PRINT CHR$(PEEK(n));
140 NEXT n
150 PRINT
160 PRINT
170 INPUT "Voulez-vous examiner d'autres
 parties dela memoire ( /N) ";f$
180 f$=UPPER$(f$)
190 IF f$<>"N" THEN GOTO 10
200 END
210 PEN effet:PRINT:PRINT TAB(12) "Entre
e incorrecte!"
220 PEN normal:GOSUB 230:RETURN
230 PRINT:PRINT TAB(7) "<Appuyez sur une
 touche S.V.P.>"
240 f$=INKEY$:IF f$="" THEN GOTO 240
250 RETURN
```

Mémoire 3

=========

En entrant ce programme vous faites d'une pierre deux coups: premièrement l'affichage à l'écran du contenu de la mémoire apparaît sous une forme plus ordonnée (adresse de début de la zone examinée, contenu de cette adresse et des 7 octets suivants avec chaque fois l'affichage CHR$ correspondant, le tout sur une ligne de l'écran) et d'autre part, seuls sont affichés les caractères CHR$ représentables (entre 32 = espace et 126 = ligne zig-zag). Les caractères non représentables ont été simplement remplacés par CHR$(46) = '.'.

Vous pouvez utiliser ce programme de deux façons: si vous entrez simplement l'adresse de début et que vous appuyez ensuite sur la touche ENTER, vous verrez uniquement le contenu des 8 octets situés à partir de cette adresse. Si par contre, vous appuyez, immédiatement après avoir appuyé sur la touche ENTER et pendant que l'affichage à l'écran est en train de se constituer, sur une touche à fonction de répétition, par exemple la touche espace, vous verrez s'afficher à l'écran le contenu des octets suivants, toujours affichés par lignes de 8.

Examinons par exemple où se trouve en mémoire notre programme BASIC Mémoire 3. Sur le CPC comme sur tous les ordinateurs, les instructions ne sont stockées que sous forme d'un code (=token) alors que les mots placés à la suite d'une instruction REM sont conservés comme ils ont été écrits. Si vous entrez par exemple 1006 comme adresse de départ, vous devriez maintenant découvrir le mot "octets". Voyez en effet la ligne 140 du listing:

```
140 REM Chaque ligne de l'écran reçoit
          8 octets
```

Ceci ne fonctionne évidemment que si vous avez bien entré le programme exactement comme il figure sur le listing, sans oublier même un espace.

Si vous entrez 1006 comme adresse de départ et que vous appuyez immédiatement sur la touche espace, vous voyez apparaître la suite de notre programme Mémoire 3.

Indication pour l'utilisation du programme 'Mémoire 3':

Comme vous avez certainement déjà remarqué, les codes correspondant au contenu des zones de mémoire que vous pouvez examiner avec notre programme ne sont pas indiqués avec le système décimal que nous connaissons bien pour l'avoir appris à l'école mais avec le système hexadécimal. Il y a à celà deux raisons: la première est que le système hexadécimal est le système par excellence des fans du langage-machine et de tous ceux qui auscultent la mémoire des ordinateurs; et c'est bien ce que nous essayons de faire! La deuxième raison est qu'on peut placer plus d'informations sur une ligne de 40 caractères avec le système hexadécimal qu'on ne le pourrait avec le système décimal (le nombre décimal de 3 chiffres '255' devient en hexadécimal un nombre de deux chiffres 'FF', de même 65535 (déc.)= FFFF (hexa.).

Si vous ne connaissez pas encore le système hexadécimal, n'ayez aucune crainte, votre CPC vous permet de convertir très aisément les nombres décimaux en nombres hexadécimaux avec l'instruction PRINT HEX$(xx). Par exemple PRINT HEX$(65535) vous donnera: FFFF.

```
10 REM Examen de la memoire 3
20 REM Examen de la memoire avec PEEK et
25 REM    CHR$ ( dans un format clair)
30 REM Programmes Basic sur le CPC464
40 REM Copyright 1984 DATA BECKER &
45 REM    Rainer Lueers
50 INK 0,1:INK 1,24:INK 2,1,24:effet=2:normal=1
60 KEY DEF 18,1
70 MODE 1
80 REM Delimiter le cadre de l'examen
90 INPUT "Entrer l'adresse de debut ";a
100 IF a<0 OR a>65535 THEN GOSUB 330:GOTO 70
110 a$=HEX$(a,4)
120 REM Les lignes suivantes permettent
121 REM    d'obtenir une sortie sur ecran
122 REM    optimale avec 40 caracteres par
123 REM    ligne
130 PRINT a$;" ";
140 REM Chaque ligne de l'ecran recoit
141 REM    8 octets
150 FOR z=0 TO 7
160 zz=PEEK(a+z):zz$=HEX$(zz,2)
170 PRINT zz$;" ";
180 NEXT z
190 REM Sortie de la chaine de
191 REM    caracteres avec un traitement
192 REM    particulier lorsque la valeur
193 REM    PEEK est <32 ou >126
200 FOR z=0 TO 7
210 zz=a+z
220 zz=PEEK(zz)
230 IF zz<32 THEN zz=46
240 IF zz>126 THEN zz=46
250 PRINT CHR$(zz);
260 NEXT z
270 PRINT
280 a=a+8
290 a$=INKEY$
300 IF a$="" THEN GOTO 90
310 GOTO 110
```

```
320 END
330 PEN effet:PRINT:PRINT TAB(12) "Entree incorrecte!"
340 PEN normal:GOSUB 350:RETURN
350 PRINT:PRINT TAB(7) "<Appuyez sur une touche S.V.P.>"
360 f$=INKEY$:IF f$="" THEN GOTO 360
370 RETURN
```

Mémoire 4

=========

Nous avons examiné jusqu'à présent la mémoire de notre ordinateur et nous n'avons trouvé au début que des nombres (Mémoire 1), puis des caractères bizarres (Mémoire 2) mais nous avons enfin obtenu un affichage formaté des nombres et des caractères représentables (Mémoire 3). En examinant avec Mémoire 3 la zone occupée par le programme BASIC, et en comparant l'affichage sur la partie gauche de l'écran à celui de la partie droite, nous avons bien vu que notre programme BASIC ne se compose en fait que de nombres qui sont ensuite convertis par le CPC en diverses lettres ou caractères.

Pour comprendre comment cela est possible, il faut savoir comment l'ordinateur stocke en mémoire les données et les programmes. Pour économiser de la place en mémoire, les instructions sont stockées, après que vous les ayez entrées avec ENTER, sous forme d'abréviations appelées tokens (voir l'annexe). C'est ainsi que l'instruction PRINT qui devrait occuper normalement cinq octets puisqu'elle se compose de 5 lettres n'occupe qu'un octet en mémoire.

Mais si vous écrivez 'PRINT' dans une ligne REM ou entre guillemets, par exemple PRINT"PRINT", le mot PRINT que l'ordinateur ne considérera pas comme une instruction mais comme du texte sera stocké lettre pour lettre et occupera bien 5 octets en mémoire.

Si vous essayez maintenant de rechercher grâce à notre programme

Mémoire 3 si un mot ou un texte quelconque se trouve dans un programme BASIC en mémoire et où il se trouve en mémoire,ceci risque de durer très longtemps jusqu'à ce que vous ayez pu examiner les 65535 octets de la mémoire. Autant chercher une aiguille dans une botte de foin!

Il faut donc procéder de façon plus systématique et c'est ce que permet le programme Mémoire 4! Vous pouvez en effet entrer un mot comprenant jusqu'à 6 lettres et vous indiquez ensuite à l'ordinateur dans quelle zone il doit chercher cette expression. Bien sûr, plus la zone à l'intérieur de laquelle vous lui dites d'effectuer cette recherche est grande et plus vous risquez d'attendre longtemps. Une fois la recherche effectuée, et si l'expression recherchée se trouve bien dans la zone que vous avez délimitée, l'ordinateur vous indique l'adresse où figure cette expression. Vous pouvez alors vous amuser à modifier avec l'instruction POKE votre programme BASIC (employez uniquement des codes compris entre 32 et 126 inclus). Par exemple:

En ligne 10 se trouve le mot 'Examen'. Faites d'abord rechercher à quelle adresse se trouve ce mot. Lancez le programme avec RUN, entrez 'Examen' (ENTER) et faites rechercher ces lettres dans toute la mémoire; entrez N (ENTER) pour répondre à la question suivante. Peu de temps après, vous voyez apparaître à l'écran: 'Examen' trouvé à l'adresse 374. Si vous tombez sur un nombre légèrement différent, ce n'est pas très grave. Appelons la valeur obtenue NOMBRE (pour nous NOMBRE=374). Voyons maintenant ce qui se trouve en mémoire à l'adresse NOMBRE: PRINT PEEK(NOMBRE) (ne tapez pas NOMBRE bien sûr mais la valeur que vous aviez obtenue). Résultat: '69'! Or que signifie le code 69: PRINT CHR$(69).

Résultat: 'E'!

Le nombre 69 correspond donc bien à la première lettre du mot recherché 'Examen'. Si vous listez la ligne 10, vous y voyez le mot 'Examen'. Si vous placez avec l'instruction POKE une autre valeur que 69 à l'adresse NOMBRE, vous allez changer le mot figurant à la ligne 10. Essayez par exemple de transformer le 'E' en 'G'. Pour savoir quel est le code de la lettre G, utilisez l'instruction PRINT ASC(xx): PRINT ASC("G") => 71. Il vous suffit donc d'entrer POKE NOMBRE,71 et 'Examen' en ligne 10 est devenu 'Gxamen'!

Voici comment vous pouvez grâce à nos petits programmes en apprendre un peu plus sur le mode de stockage en mémoire des programmes BASIC et avoir une idée peut-être plus précise de la complexité d'un ordinateur tel que le CPC. N'oubliez pas que le BASIC est le produit de nombreux développements et travaux.

Mais notre programme 'Mémoire 4' vous révèle également un autre aspect des programmes BASIC: le stockage du numéro de ligne. Lancez à nouveau le programme avec RUN, entrez le critère de recherche 'Examen' (ou 'Gxamen' si vous avez modifié le programme) et faites rechercher cette expression uniquement dans le programme en choisissant donc l'option 'P'. Vous verrez peu de temps après non seulement l'adresse correspondante mais aussi le numéro de ligne correspondant dans notre programme BASIC.

Le calcul du numéro de ligne est relativement simple, ... à condition qu'on sache comment l'effectuer. Tout programme BASIC sur le CPC commence normalement en 368. Les adresses 368 et 369

contiennent la longueur de la première ligne de notre programme ('PRINT PEEK (368) => 29. PRINT PEEK (369) => 0. Le calcul de la longueur peut être obtenu à partir de ces deux valeurs en appliquant la formule suivante: PRINT PEEK(368)+PEEK(369)*256 => 29. Donc: la première ligne BASIC (y compris les deux octets 368 et 369 qui en indiquent la longueur) est longue de 35 octets. Vous pouvez donc, grâce à cette information, calculer le début de la seconde ligne BASIC qui contient également les mêmes informations, et ainsi de suite. Or les adresses 370 et 371 contiennent le numéro de la première ligne BASIC: PRINT PEEK(370) => 10. PRINT PEEK(371) => 0. Calculez maintenant la valeur obtenue: PRINT PEEK(370)+PEEK(371)*256 => 10. Donc: le premier numéro de ligne de notre programme est 10 et cette ligne comporte 35 octets.

Ce sont exactement les mêmes calculs qu'effectue encore et toujours (à partir de la ligne 160) le CPC pour pouvoir indiquer le numéro de ligne lorsqu'il trouve l'expression que vous lui avez fait rechercher.

Mais peut-être en avez-vous à présent assez de la théorie! Passons donc au programme 'Mémoire 5'.

Quel(s) caractère(s) cherchez-vous

(longueur inferieure ou egale a 6) ? Examen

Dans quelle section de la memoire

faut-il chercher cette suite

de caracteres

(/N=dans toute la mémoire

/P=dans le programme) ? n

longue trouve a l'adresse 374

Exemple d'utilisation de Mémoire 4

```
10 REM Examen de la memoire 4
20 REM Rechercher en memoire une
21 REM    suite de caracteres
30 REM Programmes Basic sur le CPC464
40 REM Copyright 1984 DATA BECKER &
41 REM    Rainer Lueers
50 INK 0,1:INK 1,24:INK 2,1,24:effet=2:normal=1
60 MODE 1
70 REM Entree du critere de recherche
80 INPUT "Quel(s) caractere(s) recherchez-vous ?     (jusqu'a
 6 caracteres) ";f$
90 IF LEN(f$)>6 OR LEN(f$)=0 THEN GOSUB 540:GOTO 60
100 PRINT
110 REM Choisir zone a faire rechercher
120 INPUT "Dans quelle zone de la memoire doit      s'effectu
er la recherche                    ( /N=dans toute la memoire/P=
Programme) ";f1$
130 f1$=UPPER$(f1$)
140 PRINT
150 IF f1$="N" THEN a=0:e=65535:GOTO 370
160 IF f1$="P" THEN a=368:GOTO 180 ELSE GOTO 300
170 REM Recherche de la suite de
171 '      caracteres uniquement dans le
172 '      programme; si trouve ->
173 '      affichage des numeros de ligne
174 '      correspondants
180 aa=PEEK(a)+(PEEK(a+1)*256):bb=PEEK(a+2)+(PEEK(a+3)*256)
190 FOR n=a TO a+aa
200 IF CHR$(PEEK(n))=MID$(f$,1,1)THEN k=1
210 IF k=1 AND CHR$(PEEK(n+1))=MID$(f$,2,1)THEN k=2
220 IF k=2 AND CHR$(PEEK(n+2))=MID$(f$,3,1)THEN k=3
230 IF k=3 AND CHR$(PEEK(n+3))=MID$(f$,4,1)THEN k=4
240 IF k=4 AND CHR$(PEEK(n+4))=MID$(f$,5,1)THEN k=5
250 IF k=5 AND CHR$(PEEK(n+5))=MID$(f$,6,1)THEN k=6
260 IF k=LEN(f$) THEN PRINT "trouve ";f$;" (";n;") en ligne"
;bb
270 k=0
280 NEXT n
290 a=a+aa:IF aa=0 OR bb=0 THEN GOTO 470 ELSE GOTO 180
300 PRINT
```

```
310 REM Delimiter la zone a examiner
320 INPUT "Adresse de debut ";a
330 INPUT "Adresse de fin ";e
340 IF a>e OR a<0 OR e<0 OR a>65535 OR e>65535 THEN GOSUB 54
0:GOTO 320
350 PRINT
360 REM Recherche du critere de
361 '      recherche f$ dans la memoire de
362 '      l'adresse 'a' a l'adresse 'e'
370 FOR z=a TO e
380 IF CHR$(PEEK(z))=MID$(f$,1,1)THEN k=1
390 IF k=1 AND CHR$(PEEK(z+1))=MID$(f$,2,1)THEN k=2
400 IF k=2 AND CHR$(PEEK(z+2))=MID$(f$,3,1)THEN k=3
410 IF k=3 AND CHR$(PEEK(z+3))=MID$(f$,4,1)THEN k=4
420 IF k=4 AND CHR$(PEEK(z+4))=MID$(f$,5,1)THEN k=5
430 IF k=5 AND CHR$(PEEK(z+5))=MID$(f$,6,1)THEN k=6
440 IF k=LEN(f$) THEN PRINT "trouve ";f$;" a l'adresse";z
450 k=0
460 NEXT z
470 PRINT
480 PRINT
490 REM Fin du programme ou continuer
500 INPUT "Rechercher d'autres caracteres ( /N) ";f$
510 f$=UPPER$(f$)
520 IF f$<>"N" THEN GOTO 60
530 END
540 PEN effet:PRINT:PRINT TAB(12) "Entree incorrecte!"
550 PEN normal:GOSUB 560:RETURN
560 PRINT:PRINT TAB(7) "<Appuyez sur une touche S.V.P.>"
570 f$=INKEY$:IF f$="" THEN GOTO 570
580 PRINT:RETURN
```

Mémoire 5

=========

Alors que le programme Mémoire 3 représentait 8 octets par ligne (valeurs décimales e

t caractère CHR$), nous allons maintenant

avoir un programme plus clair puisqu'il affichera un octet seulement par ligne et fournira plus de renseignements sur chaque octet affiché.

Pour toutes les informations fournies, les codes décimaux et hexadécimaux sont successivement présentés:

1) Adresse décimale
2) Adresse hexadécimale
3) Contenu décimal de l'adresse
4) Contenu hexadécimal de l'adresse
5) Contenu de l'adresse (si possible) en caractère CHR$(

Ce programme vous sera très utile dans votre exploration de la mémoire de votre ordinateur car il vous évitera de devoir sans cesse convertir les nombres que vous obtenez. Ce programme vous permettra d'autre part très aisément d'apprendre à connaître le grand nombre de tokens (souvenez-vous: token=code d'une instruction BASIC) dont l'annexe de cet ouvrage vous fournit la liste.

Faisons un essai: ajoutez à votre programme la ligne 9: 9 PRINT (ENTER). Comme vous le savez, la mémoire du programme BASIC commence en 368. Entrons donc 368 comme adresse de début. La valeur finale nous est égale, pourvu qu'elle soit supérieure en

tout cas à l'adresse de début (appuyons donc simplement sur ENTER). L'affichage de la mémoire se déroule jusqu'à l'adresse de début + 2000 = 2368. Pour notre expérience, il nous suffit de faire afficher les 10 premières adresses de la mémoire, donc adresse finale 378! Listez ensuite les ligne 9 et 10 pour contrôler l'affichage.

Voici brièvement comment interpréter l'affichage du contenu de la mémoire:

Adresse 368: contient 6	(368 et 369 donnent la longueur de la ligne)
Adresse 369: contient 0	(Adr 368+(256*Adr 369) = 6 octets)
Adresse 370: contient 9	(370 et 371 donnent le numéro de la ligne)
Adresse 371: contient 0	(Adr 370+(256*Adr 371) = numéro 9)
Adresse 372: contient 191	(token de l'instruction PRINT)
Adresse 373: contient 0	(Marque de fin de ligne)
Adresse 374: contient 35	(374 et 375 donnent la longueur de la ligne)
Adresse 375: contient 0	(Adr 374+(256*Adr 375) = 29 octets)
Adresse 376: contient 10	(376 et 377 donnent le numéro de la ligne)
Adresse 377: contient 0	(Adr 376+(256*Adr 377) = numéro 10)
Adresse 378: contient 197	(token de l'instruction REM)

Intéressons-nous à l'adresse 372. Entrez POKE 372,197 (197 est le token de 'REM') et listons la ligne 9. Vous voyez que l'instruction PRINT de la ligne 9 a été remplacée par une instruction REM. Notez simplement pour le moment qu'il n'existe pas seulement des tokens à un octet mais aussi des tokens à deux

octets. Nous n'en dirons pas plus pour le moment mais vous pouvez consulter l'annexe comportant la liste des tokens.

```
10 REM Examen de la memoire 5
20 REM Sortie du contenu de la memoire
21 '      un octet par ligne
30 REM Programmes Basic pour le CPC464
40 REM Copyright 1984 DATA BECKER &
41 '      Rainer Lueers
50 INK 0,1:INK 1,24:INK 2,1,24:effet=2:normal=1
60 MODE 1
70 REM Delimiter la zone a examiner
80 INPUT "Adresse de debut ";a
90 REM
100 PRINT "Adresse de fin (Chiffre/<ENTER>->";a+2000;")":INP
UT e
110 IF a<0 OR e<0 OR a>65535 OR e>65535 THEN GOSUB 270:GOTO
60
120 IF e=0 THEN e=a+2000
130 CLS
140 REM Ordre de la sortie:
141 '      Adresse decimale
142 '      Adresse hexadecimale
143 '      Contenu de l'adresse en decimal
144 '      Contenu de l'adresse hexadecimal
145 '      Contenu de l'adresse en CHR$(
150 FOR z=a TO e
160 PRINT z;
170 PRINT TAB(8) HEX$(z,4);
180 PRINT TAB (18) PEEK(z);
190 PRINT TAB(24) HEX$(PEEK(z),2);
200 PRINT TAB(30);
210 n1=PEEK(z)
220 IF n1<32 THEN n1=46
230 IF n1>126 THEN n1=46
240 PRINT CHR$(n1)
250 NEXT z
260 END
270 PEN effet:PRINT:PRINT TAB(12) "Entree incorrecte!"
280 PEN normal:GOSUB 290:RETURN
290 PRINT:PRINT TAB(7) "<Appuyez sur une touche S.V.P.>"
300 f$=INKEY$:IF f$="" THEN GOTO 300
310 RETURN
```

Editeur graphique

=================

Le graphisme est certainement plus simple à programmer sur le CPC464 que sur beaucoup d'autres microordinateurs similaires, mais il reste néammoins relativement compliqué.

Il n'y a en effet que deux instructions graphiques directes (PLOT pour créer un point et DRAW pour dessiner une ligne) et il faut d'autre part savoir jongler avec les coordonnées si on ne veut pas risquer de ne pas créer ou dessiner exactement les points qu'on aurait souhaité.

Mais le graphisme devient un jeu d'enfant avec le programme d'éditeur graphique que nous vous présentons maintenant.

Nous avons longuement réfléchi avant de décider du mode de commande du curseur graphique qu'il convenait de choisir (il aurait été en effet inutile de réaliser un éditeur graphique, si c'était pour utiliser uniquement l'adressage absolu!). Nous avons éliminé le joystick et les touches curseur parce qu'ils ne nous permettent pas de distinguer plus de 4 directions.

Comme nous voulions avoir un bloc de 8 directions (les 4 points cardinaux plus les 4 diagonales), nous avons simplement réaffecté une grande partie du clavier de façon à disposer d'un bloc de 8 touches correspondant aux 8 directions.

Le programme "Editeur graphique" ne vous permet pas simplement de dessiner dans 8 directions:

'C' vous permet de positionner le curseur de manière absolue (par exemple 320,200 correspond au centre de l'écran).

'D' vous permet de dupliquer une zone de l'écran (vous marquez avec le premier curseur l'angle inférieur gauche de cette zone et avec le second curseur l'angle supérieur droit puis vous marquez avec le troisième curseur l'angle inférieur gauche de la zone où cette zone doit être copiée). La fonction 'OR', au contraire de la fonction 'AND' n'efface pas le fond de l'écran. Vous pouvez également faire afficher l'original agrandi deux fois.

'F' vous permet de choisir parmi les couleurs possibles.

'G' vous permet de tracer une ligne d'une longueur quelconque entre deux points.

'K' vous permet de tracer un cercle autour du curseur qui en sera le centre, après avoir entré le rayon de ce cercle.

'L' vous permet d'effacer une zone que vous définissez par deux positions du curseur (angle inférieur gauche et angle supérieur droit).

'R'vous permet de créer un rectangle que vous définissez par deux positions du curseur (angle inférieur gauche et angle supérieur droit).

'T' vous permet d'afficher un texte à partir de l'emplacement du curseur.

Vous voyez qu'il ne manque à notre éditeur graphique que la possibilité de peindre des zones, de sauvegarder l'image créée sur cassette (une sauvegarde de 16 K dure vraiment trop longtemps, attendons plutôt le lecteur de disquette) et une routine d'impression sur imprimante (hardcopy). Mais toutes les autres possibilités graphiques de votre ordinateur sont mises en oeuvre de façon très simple par cet éditeur graphique.

Bien sûr si vous voulez améliorer ce programme, vous pouvez donner le champ libre à votre imagination créatrice, sous réserve évidemment de ne pas surcharger la capacité mémoire de votre ordinateur.

```
10 REM Editeur graphique
20 REM Programmes Basic pour le CPC464
30 REM Copyright 1984 DATA BECKER &
35 REM   Rainer Lueers
40 INK 0,1:INK 1,24:INK 2,1,24:effet=2:normal=1
50 MODE 1
60 FOR n=3 TO 20:KEY DEF n,1:NEXT:SPEED KEY 50,2
70 PAPER 0:PEN normal
80 INPUT "Quel mode ecran ";f$
90 IF VAL(f$)<0 OR VAL(f$)>2 THEN GOSUB 1570:GOTO 80
100 IF VAL(f$)=0 THEN z1=4
110 IF VAL(f$)=1 THEN z1=2
120 IF VAL(f$)=2 THEN z1=1:mode2=3
130 MODE VAL(f$)
140 REM Fixation des fenetres
141 '      graphique et de texte
150 ORIGIN 0,0,0,640,400,20
160 WINDOW #1,1,40,25,25
170 a=320:b=200
180 PAPER #1,3-mode2:CLS #1
190 PAPER 0:CLG:PLOT a,b,1
200 REM Menu propose
201 '      Possibilites offertes: CDFGKLRT
202 '      C = Definir coordonnees Curseur
203 '      D = Duplication (Or/And/Agrand.)
204 '      F = Fixer la couleur
205 '      G = Dessiner une droite
210 '      K = Dessiner un cercle
211 '      L = Supprimer
212 '      R = Dessiner rectangle
213 '      T = Entrer texte
220 CLS #1:PRINT #1,a;b;"CDFGKLRT";
230 a$=INKEY$
240 IF a$="" THEN GOTO 230
250 GOSUB 1570
260 DRAW a,b
270 IF ASC(a$)>57 THEN GOSUB 290
280 GOTO 220
290 CLS #1:PRINT #1,a;b;
300 a$=UPPER$(a$)
```

```
310 IF a$="C" THEN PRINT #1,"Pl.Curseur";:zz=2
320 IF a$="D" THEN PRINT #1,"Duplicati.";:zz=3
330 IF a$="F" THEN PRINT #1,"Fix. Coul.";:zz=4
340 IF a$="G" THEN PRINT #1,"Des.Droite";:zz=5
350 IF a$="K" THEN PRINT #1,"Des.Cercle";:zz=6
360 IF a$="L" THEN PRINT #1,"Supprimer ";:zz=7
370 IF a$="R" THEN PRINT #1,"Rectangle ";:zz=8
380 IF a$="T" THEN PRINT #1,"Entr.Texte";:zz=9
390 IF zz=0 THEN RETURN
400 a$=INKEY$
410 IF a$="" THEN GOTO 400
420 IF a$=CHR$(13) THEN GOTO 440
430 RETURN
440 ON zz GOSUB 410,470,520,860,900,1000,1080,1280,1450
450 RETURN
460 REM C = Fixer curseur avec coord.
470 CLS #1:INPUT #1,"Ou(x,y)";a$,b$
480 IF VAL(a$)<0 OR VAL(a$)>640 OR VAL(a$)<20 OR VAL(a$)>400
 THEN GOTO 470
490 a=VAL(a$):b=VAL(b$):PLOT a,b
500 RETURN
510 REM D = Duplication (Or/And/Agrand.)
520 pointa=a:pointb=b:CLS #1:PRINT #1,"Fixer curseur 1"
530 a$=INKEY$
540 IF a$="" THEN GOTO 530
550 couleur=TEST(a,b):point1=a:point2=b
560 FOR m=1 TO 10:PLOT a,b,couleur:PLOT a,b,1:PLOT a,b,2:PLO
T a,b,3:PLOT a,b,1:PLOT a,b,couleur:NEXT m
570 GOSUB 1570
580 IF a$=CHR$(13) THEN a1=a:b1=b ELSE GOTO 530
590 CLS #1:PRINT #1,"Fixer curseur 2"
600 a$=INKEY$
610 IF a$="" THEN GOTO 600
620 couleur=TEST(a,b):point1=a:point2=b
630 FOR m=1 TO 10:PLOT a,b,couleur:PLOT a,b,1:PLOT a,b,2:PLO
T a,b,3:PLOT a,b,1:PLOT a,b,couleur:NEXT m
640 GOSUB 1570
650 IF a$=CHR$(13) THEN a2=a:b2=b:IF a2<=a1 OR b2<=b1 THEN G
OTO 590 ELSE GOTO 660 ELSE GOTO 600
660 CLS #1:PRINT #1,"Fixer curseur 3"
```

```
670 a$=INKEY$
680 IF a$="" THEN GOTO 670
690 couleur=TEST(a,b):point1=a:point2=b
700 FOR m=1 TO 10:PLOT a,b,couleur:PLOT a,b,1:PLOT a,b,2:PLO
T a,b,3:PLOT a,b,1:PLOT a,b,couleur:NEXT m
710 GOSUB 1570
720 IF a$=CHR$(13) THEN a3=a:b3=b ELSE GOTO 670
730 CLS #1:INPUT #1,"O(r) ou N(ot) ";f$:f$=UPPER$(f$):IF LEF
T$(f$,1)="O" THEN flag=1 ELSE flag=0
740 CLS #1:INPUT #1,"agrandi (O/ ) ";f$:f$=UPPER$(f$):IF LEF
T$(f$,1)="O" THEN flag2=1 ELSE flag2=0
750 IF flag2<>0 THEN GOTO 800
760 n1=0:m1=0:FOR n=a1 TO a2 STEP z1:FOR m=b1 TO b2 STEP 2
770 testcouleur=TEST(n,m):IF flag=0 THEN PLOT a3+n1,b3+m1,te
stcouleur ELSE IF TEST(a3+n1,b3+m1)=0 THEN PLOT a3+n1,b3+m1,
testcouleur
780 m1=m1+2:NEXT m:m1=0:n1=n1+z1:NEXT n
790 a=pointa:b=pointb:PLOT a,b,1:RETURN
800 n1=0:m1=0:FOR n=a1 TO a2 STEP z1:FOR m=b1 TO b2 STEP 2
810 ft=TEST(n,m):IF flag=0 THEN PLOT a3+n1,b3+m1,ft:PLOT a3+
n1,b3+m1+2,ft:PLOT a3+n1+z1,b3+m1,ft:PLOT a3+n1+z1,b3+m1+2,f
t
820 IF flag=1 AND TEST(a3+n1,b3+m1)=0 THEN PLOT a3+n1,b3+m1,
ft:PLOT a3+n1,b3+m1+2,ft:PLOT a3+n1+z1,b3+m1,ft:PLOT a3+n1+z
1,b3+m1+2,ft
830 m1=m1+4:NEXT m:m1=0:n1=n1+2*z1:NEXT n
840 a=pointa:b=pointb:PLOT a,b,1:RETURN
850 REM F = Fixer couleur
860 CLS #1:INPUT #1,"Quel numero ";f$
870 IF VAL(f$)<0 OR VAL(f$)>z1^2 THEN GOTO 860 ELSE PLOT a,b
,VAL(f$)
880 RETURN
890 REM G = Dessiner droite
900 CLS #1:PRINT #1,"Fixer curseur":point1=a:point2=b
910 a$=INKEY$
920 IF a$="" THEN GOTO 910
930 couleur=TEST(a,b)
940 FOR m=1 TO 10:PLOT a,b,couleur:PLOT a,b,1:PLOT a,b,2:PLO
T a,b,3:PLOT a,b,1:PLOT a,b,couleur:NEXT m
950 GOSUB 1570
```

```
960 IF a$=CHR$(13) THEN GOTO 980
970 GOTO 910
980 PLOT point1,point2,1:DRAW a,b,1:RETURN
990 REM K = Dessiner un cercle
1000 CLS #1:INPUT #1,"Rayon ";f$
1010 FOR aa=1 TO 360
1020 DEG
1030 PLOT a+VAL(f$)*COS(aa),b+VAL(f$)*SIN(aa),1
1040 NEXT aa
1050 PLOT a,b
1060 RETURN
1070 REM L = Supprimer
1080 CLS #1:PRINT #1,"Fixer curseur 1"
1090 marque1=a:marque2=b
1100 a$=INKEY$
1110 IF a$="" THEN GOTO 1100
1120 couleur=TEST(a,b):point1=a:point2=b
1130 FOR m=1 TO 10:PLOT a,b,couleur:PLOT a,b,1:PLOT a,b,2:PL
OT a,b,3:PLOT a,b,1:PLOT a,b,couleur:NEXT m
1140 GOSUB 1570
1150 IF a$=CHR$(13) THEN a1=a:b1=b ELSE GOTO 1100
1160 CLS #1:PRINT #1,"Fixer curseur 2"
1170 a$=INKEY$
1180 IF a$="" THEN GOTO 1170
1190 couleur=TEST(a,b):point1=a:point2=b
1200 FOR m=1 TO 10:PLOT a,b,couleur:PLOT a,b,1:PLOT a,b,2:PL
OT a,b,3:PLOT a,b,1:PLOT a,b,couleur:NEXT m
1210 GOSUB 1570
1220 IF a$=CHR$(13) THEN a2=a:b2=b ELSE GOTO 1170
1230 ORIGIN 0,0,a1,a2,b2,b1:CLG:
1240 ORIGIN 0,0,0,640,400,20
1250 marque1=a:marque2=b:PLOT a,b,1
1260 RETURN
1270 REM R = Dessiner Rectangle
1280 CLS #1:PRINT #1,"Fixer curseur 1"
1290 marque1=a:marque2=b
1300 a$=INKEY$
1310 IF a$="" THEN GOTO 1300
1320 couleur=TEST(a,b):point1=a:point2=b
1330 FOR m=1 TO 10:PLOT a,b,couleur:PLOT a,b,1:PLOT a,b,2:PL
```

```
OT a,b,3:PLOT a,b,1:PLOT a,b,couleur:NEXT m
1340 GOSUB 1570
1350 IF a$=CHR$(13) THEN a1=a:b1=b ELSE GOTO 1300
1360 CLS #1:PRINT #1,"Fixer curseur 2"
1370 a$=INKEY$
1380 IF a$="" THEN GOTO 1370
1390 couleur=TEST(a,b):point1=a:point2=b
1400 FOR m=1 TO 10:PLOT a,b,couleur:PLOT a,b,1:PLOT a,b,2:PL
OT a,b,3:PLOT a,b,1:PLOT a,b,couleur:NEXT m
1410 GOSUB 1570
1420 IF a$=CHR$(13) THEN a2=a:b2=b ELSE GOTO 1370
1430 PLOT a1,b1:DRAW a2,b1,1:DRAW a2,b2,1:DRAW a1,b2,1:DRAW
a1,b1,1:a=marque1:b=marque2:PLOT a,b,1:RETURN
1440 REM T = Entrer texte
1450 CLS #1:PRINT #1,"Fixer curseur"
1455 marque1=a:marque2=b
1460 a$=INKEY$
1470 IF a$="" THEN GOTO 1460
1480 couleur=TEST(a,b):point1=a:point2=b
1485 FOR m=1 TO 10:PLOT a,b,couleur:PLOT a,b,1:PLOT a,b,2:PL
OT a,b,3:PLOT a,b,1:PLOT a,b,couleur:NEXT m
1490 GOSUB 1570
1500 IF a$=CHR$(13) THEN a1=a:b1=b:PLOT a1,b1,1 ELSE GOTO 14
60
1510 TAG
1520 CLS #1:INPUT #1,"Texte ";a$
1530 PRINT a$;
1540 a=marque1:b=marque2:PLOT a,b,1
1550 TAGOFF:RETURN
1560 REM Calcul des mouvements du curseur (8 directions)
1570 IF ASC(a$)=54 THEN a=a+z1
1580 IF ASC(a$)=57 THEN a=a+z1:b=b+z1
1590 IF ASC(a$)=56 THEN b=b+2
1600 IF ASC(a$)=55 THEN a=a-z1:b=b+z1
1610 IF ASC(a$)=52 THEN a=a-z1
1620 IF ASC(a$)=49 THEN a=a-z1:b=b-z1
1630 IF ASC(a$)=50 THEN b=b-2
1640 IF ASC(a$)=51 THEN a=a+z1:b=b-z1
1650 RETURN
1660 a$=INKEY$
```

```
1670 IF a$="" THEN 1660
1680 PRINT ASC(a$)
1690 GOTO 1660
```

Editeur de son

==============

Le CPC dispose d'un processeur sonore vraiment remarquable mais plus d'un fana de musique aura certainement déjà été découragé par le complexité des instructions musicales et l'apparent illogisme de leur syntaxe. Comment s'expliquer en effet qu'il faille appeler avec l'instruction SOUND les canaux des notes en premier lieu et n'appeler le volume qu'en quatrième position? Comment se rappeler précisément des diverses possibilités des techniques de rendez-vous?

Nous ne pensons pas que l'apprentissage par coeur puisse être ici d'un grand secours mais plutôt que la seule solution est de rechercher sans cesse dans le manuel les informations dont on a besoin.

Ou encore: comment se rappeler quelle courbe d'enveloppe ou de volume est actuellement activée et quelle courbe nous pouvons activer? Faut-il faire des petites notes sur papier?

Notre 'éditeur sonore' vous offre une solution bien plus commode. Vous trouverez en effet tous les paramètres avec leur signification en français sur une page de l'écran. Vous pourrez ainsi, simplement en appuyant simultanément sur la touche 'SHIFT' et sur une autre touche, monter (par exemple <SHIFT> 'H' => augmente la hauteur de la note) ou en appuyant simultanément sur la touche 'CTRL' et sur une autre touche, descendre (par exemple <CTRL> 'L' => diminue le volume).

Vous pouvez ainsi produire à l'écran, très tranquillement, n'importe quelle note ou bruit en appuyant ensuite sur la touche ENTER pour activer le son ainsi édité. N'oubliez pas qu'il faut activer, pour produire un son avec notre éditeur sonore, le volume, la hauteur de la note et au moins un canal sonore.

Vous pouvez activer jusqu'à 16 courbes différentes pour le son d'une part et pour le volume d'autre part et vous pouvez utiliser ces courbes lors de l'édition des notes. En appuyant sur la touche SHIFT et la touche 'N' vous pouvez entrer un numéro ENV et les paramètres correspondants. Le numéro ENV est utilisé pour la sauvegarde et n'a donc pas la même signification que le numéro de la courbe d'enveloppe. Le nombre de paramètres minimum que vous devez entrer est donc dans notre programme non pas de 3 mais de 4.

Vous pouvez entrer les courbes ENT (courbes de hauteur de la note) de la même façon en appuyant simultanément sur les touches 'SHIFT' et 'O'. Vous pouvez consulter les courbes que vous avez réalisées avec <SHIFT> 'P'.

Nous espérons que vous aurez beauxoup de plaisir à éditer et créer de nouveaux sons sur votre CPC!

```
10 REM Programme sonore
20 REM Programmes Basic pour le CPC464
30 REM Copyright 1984 DATA BECKER &
35 REM   Rainer Lueers
40 INK 0,1:INK 1,24:INK 2,1,24
50 DIM a(15,15),a1(15,15)
60 MODE 1
70 SPEED INK 20,20
80 PEN 1
90 REM Creation image ecran
100 PRINT"   Le ";:PEN 2:PRINT"son";:PEN 1:PRINT" sur le CPC
464"
110 PRINT
120 PRINT"     <SHIFT> -> +   <CTRL> -> -"
130 PRINT
140 PRINT TAB(11) "A:Canal A ";:IF a=0 THEN PRINT"et." ELSE
PEN 2:PRINT"all":PEN 1
150 PRINT TAB(11) "B:Canal B ";:IF a=0 THEN PRINT"et." ELSE
PEN 2:PRINT"all":PEN 1
160 PRINT TAB(11) "C:Canal C ";:IF a=0 THEN PRINT"et." ELSE
PEN 2:PRINT"all":PEN 1
170 PRINT
180 PRINT TAB(4) "D:Rendezvous avec canal A ";:IF d=0 THEN P
RINT"et." ELSE PEN 2:PRINT"all":PEN 1
190 PRINT TAB(4) "E:Rendezvous avec canal B ";:IF e=0 THEN P
RINT"et." ELSE PEN 2:PRINT"all":PEN 1
200 PRINT TAB(4) "F:Rendezvous avec canal C ";:IF f=0 THEN P
RINT"et." ELSE PEN 2:PRINT"all":PEN 1
210 PRINT
220 PRINT TAB(7) "G:Periode bruit";:IF g=0 THEN LOCATE 27,13
:PRINT "et." ELSE PEN 2:LOCATE 27,13:PRINT g:PEN 1
230 PRINT
240 PRINT TAB(4) "H:Hauteur note ";:IF h=0 THEN LOCATE 22,15
:PRINT"et." ELSE PEN 2:LOCATE 22,15:PRINT h:PEN 1
250 PRINT
260 PRINT TAB(4) "I:Duree en 1/100 Sec.    ";:IF i=0 THEN PR
INT"et." ELSE PEN 2:LOCATE 28,21:PRINT i:PEN 1
270 PRINT
280 PRINT TAB(4) "J:Appel d'une courbe ENV ";:IF j=0 THEN LO
CATE 33,19:PRINT"0" ELSE PEN 2:LOCATE 32,19:PRINT j:PEN 1
```

```
290 PRINT TAB(4) "K:Appel d'une courbe ENT ";:IF k=0 THEN LO
CATE 33,20:PRINT"0" ELSE PEN 2:LOCATE 32,20:PRINT j:PEN 1
300 PRINT
310 PRINT TAB(11) "L:Volume ";:IF l=0 THEN LOCATE 25,22:PRIN
T"et." ELSE PEN 2:LOCATE 25,22:PRINT l:PEN 1
320 PRINT
330 PRINT TAB(4) "N:Entr. ENV  O:Entr. ENT  P:Liste"
340 a$=INKEY$
350 REM Fixer canal son et rendez-vous
360 a1=a+b+c+d+e+f
370 REM Interrogation clavier et
375 REM   production son lorsque <ENTER>
376 REM   enfonce
380 IF a$=CHR$(13) THEN SOUND a1,h,i,l,j,k,g
390 IF a$=CHR$(14) OR a$=CHR$(78) THEN GOTO 2120
400 IF a$=CHR$(15) OR a$=CHR$(79) THEN GOTO 2250
410 IF a$=CHR$(16) OR a$=CHR$(80) THEN GOTO 2370
420 IF a$="" THEN GOTO 340
430 IF ASC(a$)>12 AND ASC(a$)<65 THEN GOTO 340
440 IF ASC(a$)>76 THEN GOTO 340
450 IF ASC(a$)>12 THEN GOTO 1250
460 REM Modification de l'ecran quand
462 REM   une touche a ete actionnee
470 ON ASC(a$) GOSUB 490,530,570,610,650,690,730,820,910,100
0,1080,1160
480 GOTO 340
490 a=0
500 LOCATE 21,5
510 PRINT"et."
520 RETURN
530 b=0
540 LOCATE 21,6
550 PRINT"et."
560 RETURN
570 c=0
580 LOCATE 21,7
590 PRINT"et."
600 RETURN
610 d=0
620 LOCATE 30,9
```

```
630 PRINT"et."
640 RETURN
650 e=0
660 LOCATE 30,10
670 PRINT"et."
680 RETURN
690 f=0
700 LOCATE 30,11
710 PRINT"et."
720 RETURN
730 IF g>0 THEN g=g-1
740 IF g=0 THEN LOCATE 27,13:PRINT"et.":RETURN
750 LOCATE 27,13
760 PRINT"   "
770 LOCATE 27,13
780 PEN 2
790 PRINT g
800 PEN 1
810 RETURN
820 IF h>0 THEN h=h-1
830 IF h=0 THEN LOCATE 22,15:PRINT"et.":RETURN
840 LOCATE 22,15
850 PRINT"    "
860 LOCATE 22,15
870 PEN 2
880 PRINT h
890 PEN 1
900 RETURN
910 IF i>0 THEN i=i-1
920 IF i=0 THEN LOCATE 28,17:PRINT"et.":RETURN
930 LOCATE 28,17
940 PRINT"      "
950 LOCATE 28,17
960 PEN 2
970 PRINT i
980 PEN 1
990 RETURN
1000 IF j>0 THEN j=j-1
1010 LOCATE 32,19
1020 PRINT"  "
```

```
1030 LOCATE 32,19
1040 IF j>0 THEN PEN 2
1050 PRINT j
1060 PEN 1
1070 RETURN
1080 IF k>0 THEN k=k-1
1090 LOCATE 32,20
1100 PRINT"  "
1110 LOCATE 32,20
1120 IF k>0 THEN PEN 2
1130 PRINT k
1140 PEN 1
1150 RETURN
1160 IF l>0 THEN l=l-1
1170 LOCATE 25,22
1180 PRINT"   "
1190 LOCATE 25,22
1200 IF l=0 THEN PRINT"et.":RETURN
1210 PEN 2
1220 PRINT l
1230 PEN 1
1240 RETURN
1250 ON (ASC(a$)-64) GOSUB 1270,1330,1390,1450,1510,1570,163
0,1710,1790,1870,1950,2030
1260 GOTO 340
1270 a=1
1280 LOCATE 21,5
1290 PEN 2
1300 PRINT"all"
1310 PEN 1
1320 RETURN
1330 b=2
1340 LOCATE 21,6
1350 PEN 2
1360 PRINT"all"
1370 PEN 1
1380 RETURN
1390 c=4
1400 LOCATE 21,7
1410 PEN 2
```

```
1420 PRINT"all"
1430 PEN 1
1440 RETURN
1450 d=8
1460 LOCATE 30,9
1470 PEN 2
1480 PRINT"all"
1490 PEN 1
1500 RETURN
1510 e=16
1520 LOCATE 30,10
1530 PEN 2
1540 PRINT"all"
1550 PEN 1
1560 RETURN
1570 f=32
1580 LOCATE 30,11
1590 PEN 2
1600 PRINT"all"
1610 PEN 1
1620 RETURN
1630 IF g<15 THEN g=g+1
1640 LOCATE 27,13
1650 PRINT"  "
1660 LOCATE 27,13
1670 PEN 2
1680 PRINT g
1690 PEN 1
1700 RETURN
1710 IF h<4095 THEN h=h+1
1720 LOCATE 22,15
1730 PRINT"    "
1740 LOCATE 22,15
1750 PEN 2
1760 PRINT h
1770 PEN 1
1780 RETURN
1790 IF i<32767 THEN i=i+1
1800 LOCATE 28,17
1810 PRINT"     "
```

```
1820 LOCATE 28,17
1830 PEN 2
1840 PRINT i
1850 PEN 1
1860 RETURN
1870 IF j<15 THEN j=j+1
1880 LOCATE 32,19
1890 PRINT"  "
1900 LOCATE 32,19
1910 PEN 2
1920 PRINT j
1930 PEN 1
1940 RETURN
1950 IF k<15 THEN k=k+1
1960 LOCATE 32,20
1970 PRINT"  "
1980 LOCATE 32,20
1990 PEN 2
2000 PRINT k
2010 PEN 1
2020 RETURN
2030 IF l<15 THEN l=l+1
2040 LOCATE 25,22
2050 PRINT"   "
2060 LOCATE 25,22
2070 PEN 2
2080 PRINT l
2090 PEN 1
2100 RETURN
2110 REM Creation d'une nouvelle courbe ENV
2120 LOCATE 4,24
2130 PRINT"                                    "
2140 LOCATE 4,24
2150 INPUT"No ENV et nombre de parametres";z,z1
2160 IF z<0 OR z>15 OR (z1<>4 AND z1<>7 AND z1<> 10 AND z1<>
13 AND z1<>16) THEN GOTO 2120 ELSE LOCATE 4,24:PRINT"
                                 ":LOCATE 4,24
2170 IF z1=4 THEN INPUT"4 Par. ";a(z,0),a(z,1),a(z,2),a(z,3)
2180 IF z1=7 THEN INPUT"7 Par. ";a(z,0),a(z,1),a(z,2),a(z,3)
,a(z,4),a(z,5),a(z,6)
```

```
2190 IF z1=10 THEN INPUT"10 Par. ";a(z,0),a(z,1),a(z,2),a(z,
3),a(z,4),a(z,5),a(z,6),a(z,7),a(z,8),a(z,9)
2200 IF z1=13 THEN INPUT"13 Par. ";a(z,0),a(z,1),a(z,2),a(z,
3),a(z,4),a(z,5),a(z,6),a(z,7),a(z,8),a(z,9),a(z,10),a(z,11)
,a(z,12)
2210 IF z1=16 THEN INPUT"16 Par. ";a(z,0),a(z,1),a(z,2),a(z,
3),a(z,4),a(z,5),a(z,6),a(z,7),a(z,8),a(z,9),a(z,10),a(z,11)
,a(z,12),a(z,13),a(z,14),a(z,15)
2220 ENV a(z,0),a(z,1),a(z,2),a(z,3),a(z,4),a(z,5),a(z,6),a(
z,7),a(z,8),a(z,9),a(z,10),a(z,11),a(z,12),a(z,13),a(z,14),a
(z,15)
2230 GOTO 2370
2240 REM Creation d'une nouvelle courbe ENT
2250 LOCATE 4,24
2260 PRINT "                                    "
2270 LOCATE 4,24
2280 INPUT"No ENT et nombre de parametres";z,z1
2290 IF z<0 OR z>15 OR (z1<>4 AND z1<>7 AND z1<> 10 AND z1<>
13 AND z1<>16) THEN GOTO 2250 ELSE LOCATE 4,24:PRINT"
                                    ":LOCATE 4,24
2300 IF z1=4 THEN INPUT"4 Par. ";a1(z,0),a1(z,1),a1(z,2),a1(
z,3)
2310 IF z1=7 THEN INPUT"7 Par. ";a1(z,0),a1(z,1),a1(z,2),a1(
z,3),a1(z,4),a1(z,5),a1(z,6)
2320 IF z1=10 THEN INPUT"10 Par. ";a1(z,0),a1(z,1),a1(z,2),a
1(z,3),a1(z,4),a1(z,5),a1(z,6),a1(z,7),a1(z,8),a1(z,9)
2330 IF z1=13 THEN INPUT"13 Par. ";a1(z,0),a1(z,1),a1(z,2),a
1(z,3),a1(z,4),a1(z,5),a1(z,6),a1(z,7),a1(z,8),a1(z,9),a1(z,
10),a1(z,11),a1(z,12)
2340 IF z1=16 THEN INPUT"16 Par. ";a1(z,0),a1(z,1),a1(z,2),a
1(z,3),a1(z,4),a1(z,5),a1(z,6),a1(z,7),a1(z,8),a1(z,9),a1(z,
10),a1(z,11),a1(z,12),a1(z,13),a1(z,14),a1(z,15)
2350 ENT a1(z,0),a1(z,1),a1(z,2),a1(z,3),a1(z,4),a1(z,5),a1(
z,6),a1(z,7),a1(z,8),a1(z,9),a1(z,10),a1(z,11),a1(z,12),a1(z
,13),a1(z,14),a1(z,15)
2360 REM Representation sur l'ecran
2362 '     des courbes ENV et ENT
2364 '     existantes
2370 MODE 1
2380 a$=""
```

```
2390 INPUT"ENV-Liste ou ENT-Liste (V/T) ";a$
2400 IF a$="" THEN GOTO 60
2410 IF a$<>"V" AND a$<>"v" AND a$<>"T" AND a$<>"t" THEN GOT
0 2370
2420 IF a$<>"v" AND a$<>"V" THEN GOTO 2530
2430 REM Sortie des courbes ENV
2440 FOR n=0 TO 15
2450 FOR m=0 TO 15
2460 IF a(n,0)=0 THEN GOTO 2470 ELSE PRINT a(n,m);
2470 NEXT m
2480 PRINT:PAPER 3:PRINT n;".";:PAPER 4
2490 NEXT n
2500 a$=INKEY$
2510 IF a$="" THEN GOTO 2500 ELSE GOTO 60
2520 REM Sortie des courbes ENT
2530 FOR n=0 TO 15
2540 FOR m=0 TO 15
2550 IF a1(n,0)=0 THEN GOTO 2560 ELSE PRINT a1(n,m);
2560 NEXT m
2570 PRINT:PAPER 3:PRINT n;".";:PAPER 4
2580 NEXT n
2590 a$=INKEY$
2600 IF a$="" THEN GOTO 2590 ELSE GOTO 60
```

Editeur de texte
================

Nous n'avons volontairement pas appelé ce programme 'traitement de texte' car nous savions qu'en employant ce terme nous aurions nécessairement déçu l'attente des lecteurs ou programmeurs experts.

Un programme de traitement de texte doit en effet au moins permettre d'insérer des paragraphes supplémentaires dans un texte déjà existant, ce qui est cependant très difficile à réaliser si on ne dispose que de la sauvegarde sur cassette.

Un programme de traitement de texte devrait normalement permettre également un couplage du texte avec un fichier d'adresses (mailing) et permettre également une sauvegarde de section de texte indépendamment de la mémoire disponible, les diverses sections de texte ainsi sauvegardées devraient également pouvoir être affichées à l'écran lorsqu'on les "feuillette".

La sauvegarde sur cassette rend un tel programme de traitement de texte très difficile à réaliser et de toute façon un programme aussi performant et complet dépasserait largement le cadre de cet ouvrage.

Maintenant que nous avons limité votre attente, voici quelles sont les possibilités malgré tout très intéressantes de notre éditeur de texte:

- Vous pouvez stocker jusqu'à 10 pages de 22 lignes dans votre CPC

- Vous pouvez feuilleter votre texte vers le début ou vers la fin du texte en employant les commandes curseur habituelles.
- Vous pouvez écrire des textes ou les placer où vous pensez qu'ils doivent figurer.
- Vous pouvez supprimer des lettres ou des groupes de caractères.
- Si vous avez erré à l'aventure avec le curseur, vous pouvez le replacer dans son emplacement de départ simplement en appuyant sur la touche COPY.
- L'emplacement du curseur est indiqué dans une fenêtre explicative et sa position horizontale est en outre marquée nettement par un trait. Ce "trait de soulignage" est produit dans la fenêtre explicative en mode de réécriture (PRINT CHR$(22);CHR$(1)).
- Vous pouvez feuilleter votre texte vers le début en plaçant le curseur en ligne 1 et en appuyant sur la touche de curseur haut.
- Vous pouvez feuilleter votre texte vers la fin soit en plaçant le curseur en ligne 22 et en appuyant sur la touche de curseur bas, soit en appuyant simultanément sur les touches 'CTRL' et 'B'.
- Vous pouvez sauvegarder votre texte en appuyant simultanément sur les touches CTRL et S puis en entrant le nom de fichier souhaité.
- Vous pouvez charger un texte en appuyant simultanément sur les touches CTRL et L.
- Vous pouvez enfin faire rechercher certaines expressions dans votre texte (et faire poursuivre la recherche après qu'une expression ait été trouvée) et faire imprimer votre texte.

La dernière fonction, l'impression du texte que vous avez entré, mérite une explication supplémentaire. Vous avez entré votre texte

dans un format de 40 colonnes. Mais lors de l'impression, votre CPC éditera le texte sur l'imprimante avec le nombre de caractères par ligne que vous choisissez, le minimum étant 40 caractères par ligne.

Encore un conseil:

Si vous ne possédez pas d'imprimante, vous pouvez quand même vous amuser à faire éditer le texte adapté pour l'impression sur l'écran du moniteur, en mode 80 colonnes. Mais vous devez pour celà apporter de légères modifications au programme:

1) Ajoutez la ligne suivante:

```
755 MODE 2
```

2) Remplacez en lignes 780 et 790 les instructions PRINT #8 par de simples PRINT.

3) Il est aussi préférable que le programme attende un peu après avoir édité le texte. Sur l'imprimante, ce n'est pas nécessaire puisque vous pouvez toujours consulter le résultat obtenu sur papier. Vous pourriez donc rajouter en ligne 810 soit une boucle (FOR n=1 TO 10000) avant l'instruction de retour au mode 1 'MODE 1', soit même un message vous demandant si vous voulez continuer (PRINT"FRAPPEZ UN TOUCHE") suivi d'une interrogation du clavier avec INKEY$.

Vous pouvez maintenant faire diverses expériences et voir par exemple comment le texte se présente en 80, 70 ou 60 colonnes.

```
10 REM Editeur de texte
20 REM Programmes Basic pour le CPC464
30 REM Copyright 1984 DATA BECKER &
31 REM   Rainer Lueers
40 ON ERROR GOTO 850
50 MODE 1:DIM a$(10,23):page=1:s1=1
60 FOR n=1 TO 10
70 FOR m=1 TO 23
80 a$(n,m)=STRING$(40," ")
90 NEXT m,n
100 REM Creation du premier ecran
110 WINDOW #1,2,39,1,1
120 WINDOW #2,1,40,2,23
130 WINDOW #3,2,39,24,24
140 WINDOW #4,1,40,25,25
150 PAPER #1,3
160 PAPER #3,3
170 PAPER #4,3
180 CLS #1:CLS #2:CLS #3:CLS #4
190 PRINT #1,STRING$(38,CHR$(94));
200 PRINT #3,STRING$(38,CHR$(241));
210 CLS #4:PRINT #4,VPOS(#2);POS(#2)
220 LOCATE #4,15,1:PRINT #4,"<CTRL> + SLRIF";:LOCATE #4,POS(
#2),1:PRINT #4,CHR$(22);CHR$(1);CHR$(95);CHR$(22);CHR$(0);
230 IF s1<page THEN s1=page
240 REM Attendre la frappe d'une touche
250 a$=INKEY$
260 IF a$="" THEN GOTO 250
270 GOTO 630
280 GOSUB 330
290 GOSUB 550
300 GOSUB 570
310 PRINT #2,a$;:IF a$=CHR$(241) THEN PRINT #2
320 GOTO 210
330 REM Ediieur ecran:
331 '     reconaissance des limites,
332 '     possibilite de feuilleter et
333 '     touche <DEL>
340 IF ASC(a$)<244 AND ASC(a$)>240 AND marque<>1 THEN marque
=1:v=VPOS(#2):p=POS(#2)
```

```
350 REM CHR$(224)
351 '     c'est-a-dire: 'COPY' enfoncee
352 '     signifie: ramener curseur au
353 '     point de depart
360 IF ASC(a$)=224 AND marque=1 THEN marque=0:LOCATE #2,p,v:
a$="":GOTO 540
370 IF POS(#2)=40 AND VPOS(#2)=22 AND a$<>CHR$(13) AND ASC(a
$)<128 THEN PRINT CHR$(7);:LOCATE #2,1,22:RETURN
380 REM CHR$(127)
381 '     c'est-a-dire: 'DEL' enfoncee
382 '     signifie: supprimer caractere
383 '     a gauche du curseur
390 IF ASC(a$)=127 AND VPOS(#2)<>1 AND POS(#2)=1 THEN LOCATE
 #2,40,VPOS(#2)-1:PRINT #2," ";:LOCATE #2,40,VPOS(#2)-1:a$="
":GOTO 540
400 IF ASC(a$)=127 AND POS(#2)<>1 THEN LOCATE #2,POS(#2)-1,V
POS(#2):PRINT #2," ";:LOCATE #2,POS(#2)-1,VPOS(#2):a$="":GOT
O 540
410 REM CHR$(240)
411 '     c'est-a-dire: fleche 'haut'
412 '     enfoncee signifie: curseur
413 '     une ligne ou une page plus haut
420 IF ASC(a$)=240 AND VPOS(#2)<>1 THEN LOCATE #2,POS(#2),VP
OS(#2)-1:a$="":GOTO 540 ELSE IF ASC(a$)=240 AND VPOS(#2)=1 T
HEN IF pagee<>1 THEN page=page-1:CLS #2:FOR n=1 TO 22:PRINT
#2,a$(page,n);:NEXT n:LOCATE #2,1,1
430 REM CHR$(241)
431 '     c'est-a-dire: fleche 'bas'
432 '     enfoncee
433 '     signifie: curseur une ligne ou
434 '     une page plus bas
440 IF ASC(a$)=241 AND VPOS(#2)<>22 THEN LOCATE #2,POS(#2),V
POS(#2)+1:a$="":GOTO 540 ELSE IF ASC(a$)=241 AND VPOS(#2)=22
 THEN page=page+1:CLS #2:FOR n=1 TO 22:PRINT #2,a$(page,n);:
NEXT n:LOCATE #2,1,1
450 REM CHR$(242)
451 '     c'est-a-dire: fleche 'gauche'
452 '     enfoncee
453 '     signifie: curseur une case vers
454 '     la gauche
```

```
460 IF ASC(a$)=242 AND VPOS(#2)<>1 AND POS(#2)=1 THEN LOCATE
 #2,VPOS(#2)-1,40:a$="":GOTO 540
470 IF ASC(a$)=242 AND VPOS(#2)<>1 THEN LOCATE #2,POS(#2)-1,
VPOS(#2):a$="":GOTO 540
480 REM CHR$(243)
481 '     c'est-a-dire: fleche 'droite'
482 '     enfoncee
483 '     signifie: curseur une case vers
484 '     la droite
490 IF ASC(a$)=243 AND VPOS(#2)<>22 AND POS(#2)=40 THEN LOCA
TE #2,1,VPOS(#2)+1:a$="":GOTO 540
500 IF ASC(a$)=243 AND VPOS(#2)<>22 THEN LOCATE #2,POS(#2)+1
,VPOS(#2):a$="":GOTO 540
510 IF ASC(a$)<244 AND ASC(a$)>239 THEN marque=0:a$="":GOTO
540
520 IF ASC(a$)=224 THEN a$=""
530 IF a$=CHR$(13) THEN a$=CHR$(241)
540 RETURN
550 IF VPOS(#2)=22 AND a$=CHR$(241) THEN PRINT #2,a$;:MID$(a
$(page,VPOS(#2)),POS(#2)-1,1)=a$:page=page+1:a$="":CLS #2
560 RETURN
570 LOCATE #1,15,1:PRINT #1," Page";page;
580 LOCATE #3,15,1:PRINT #3," Page";page+1;
590 IF a$<>"" THEN MID$(a$(page,VPOS(#2)),POS(#2),1)=a$
600 RETURN
610 REM Tester quelle est la touche enfoncee
620 REM <CTRL> 'F'
621 '     Feuilleter
630 IF ASC(a$)=6 AND page<>10 THEN page=page+1:CLS #2:FOR n=
1 TO 22:PRINT #2,a$(page,n);:NEXT n:LOCATE #2,1,1:GOTO 300
640 REM <CTRL> 'S'
641 '     Sauvegarde sur disquette
650 IF ASC(a$)=19 THEN CLS #4:INPUT #4,"Nom ";f$:IF f$="" TH
EN GOTO 820 ELSE f$="!"+f$:OPENOUT f$:PRINT #9,s1:FOR n=1 TO
 s1:FOR m=1 TO 22:PRINT #9,a$(n,m):NEXT m,n:CLOSEOUT:GOTO 71
0
660 REM <CTRL> 'L'
661 '     Charger a partir de la
662 '     disquette
670 IF ASC(a$)=12 THEN CLS #4:INPUT #4,"Nom ";f$:IF f$="" TH
```

```
EN GOTO 820 ELSE f$="!"+f$:OPENIN f$:INPUT #9,s1:FOR n=1 TO
s1:FOR m=1 TO 22:INPUT #9,a$(n,m):NEXT m,n:CLOSEIN:page=1:CL
S #2:FOR n=1 TO 22:PRINT #2,a$(page,n);:NEXT n:GOTO 300
680 REM <CTRL> 'R'
681 '     Recherche
690 IF ASC(a$)=18 THEN CLS #4:INPUT #4,"Terme a rechercher "
;f$:IF f$="" THEN GOTO 710 ELSE FOR n=1 TO s1:FOR m=1 TO 22:
IF INSTR(a$(n,m),f$)>0 THEN CLS #2:FOR mm=1 TO 22:PRINT #2,a
$(n,mm);:NEXT mm:page=n:LOCATE #2,1,m:GOTO 710 ELSE NEXT m,n
700 REM Si le terme recherche est trouve
701 '     la recherche peut etre
702 '     poursuivie
705 O$="O"
710 IF ASC(a$)=18 THEN CLS#4:PRINT#4,n;m;:INPUT#4,"Encore(O/
)";ff$:ff$=UPPER$(ff$):IF ff$=O$THEN FOR n=page TO s1:FOR m=
VPOS(#2)+1 TO 23:IF INSTR(a$(n,m),f$)>0 THEN CLS#2:FOR mm=1
TO 22:PRINT#2,a$(n,mm);:NEXT mm:page=n:LOCATE#2,1,m:GOTO 710
 ELSE NEXT m,n
720 REM <CTRL> 'I'
721 '     Routine d'impression en 80
722 '     colonnes
730 IF ASC(a$)=9 THEN CLS#4 ELSE GOTO 820
740 INPUT #4,"Imprimante connectee (O/ ) ";ff$:ff$=UPPER$(ff
$):IF LEFT$(ff$,1)<>"O" THEN GOTO 820
750 CLS#4:INPUT #4,"Caracteres par ligne (>=40) ";z1:IF z1<4
0 THEN GOTO 750
760 a$="":FOR z2=1 TO s1:FOR z3=1 TO 22:FOR z4=1 TO 40
770 IF MID$(a$(z2,z3),z4,1)=" " THEN z5=LEN(a$) ELSE IF MID$
(a$(z2,z3),z4,1)=CHR$(241) THEN z4=40:GOTO 790
780 a$=a$+MID$(a$(z2,z3),z4,1):IF LEN(a$)=z1 THEN PRINT #8,L
EFT$(a$,z5):a$=RIGHT$(a$,LEN(a$)-z5):GOTO 800 ELSE GOTO 800
790 PRINT #8,a$:a$=""
800 IF LEFT$(a$,1)=" " THEN a$=RIGHT$(a$,LEN(a$)-1)
810 NEXT z4,z3,z2:MODE 1:GOTO 100
820 IF ASC(a$)<32 AND ASC(a$)<>13 THEN a$=""
830 GOTO 280
840 REM Routine de traitement des erreurs
850 a$=" ":RESUME NEXT
860 a$=a$+MID$(a$(z2,z3),z4,1):IF (LEN(a$)=z1 AND MID$(a$(z2
,z3),z4,1)<>" ") THEN PRINT LEFT$(a$,z5):a$=RIGHT$(a$,LEN(a$
```

```
)-z5):GOTO 800 ELSE IF RIGHT$(a$,1)=CHR$(241) THEN a$=LEFT$(
a$,LEN(a$)-1):z4=40  ELSE GOTO 800
```

Jeu de caractères français

========================

Si vous placez votre CPC moderne à côté d'une bonne vieille machine à écrire traditionnelle, vous ferez vite un constat agaçant. Vous voyez en effet d'un côté un ordinateur international (ou plutôt anglo-américain) et qui ne dispose donc pas d'accents français ni de la cédille; et de l'autre côté une machine à écrire qui possède bien toutes les lettres dont on a besoin pour écrire en français mais qui -en principe- n'est pas un ordinateur.

Notre programme a donc pour but de hisser votre CPC au niveau de la machine à écrire, au moins en ce qui concerne le clavier: le 'z' se trouve là où il doit être, à côté du 'a' et d'une manière générale toutes les touches se trouvent là où elles figurent sur les machines à écrire française normales. Vous pourrez en outre utiliser les accents et la cédille.

Mais que deviendront les lettres servant à désigner les différentes touches de votre ordinateur? Bien entendu, à moins que vous ne colliez de petites étiquettes sur les touches du clavier, les touches resteront comme avant.

Et que deviendront les caractères que vous pouvez normalement produire avec les touches qui ont maintenant reçu une nouvelle affectation? Une phrase d'exemple en ligne 250 vous permet de voir quels sont les changements effectués. Si vous faites lister cette ligne avant de lancer le programme, vous pourrvez voir quels signes ont été remplacés par des accents ou la cédille.

Si vous en avez assez d'utiliser ce jeu de caractères français,

vous pouvez éteindre votre ordinateur ou provoquer un RESET avec les touches SHIFT, CTRL ou ESC.

```
10 REM Jeu de caracteres francais
20 REM Programmes Basic pour le CPC464
30 REM Copyright 1984 DATA BECKER &
31 REM      Rainer Lueers
40 SYMBOL AFTER 32
50 REM Lecture de la forme des
51 REM   caracteres francais en lignes
52 REM   DATA
60 FOR m=1 TO 10
70 READ a
80 FOR n=1 TO 8
90 READ a(n)
100 NEXT n
110 SYMBOL a,a(1),a(2),a(3),a(4),a(5),a(6),a(7),a(8)
120 NEXT m
130 REM Nouveau jeu de caracteres
131 REM   FRANCAIS
140 KEY DEF 67,1,97,65
150 KEY DEF 59,1,122,90
160 KEY DEF 69,1,113,81
170 KEY DEF 29,1,109,77
180 KEY DEF 71,1,119,87
190 KEY DEF 38,1,58,42
200 KEY DEF 57,1,51,92
210 KEY DEF 24,1,94,126
220 KEY DEF 22,1,64
230 MODE 1
240 REM Phrase d'exemple
250 PRINT"Le probl°me ou plutùt l'intçr§t qui na`tde toute v
olontç de crçer des textes    fran¨ais sur un ordinateur est
 le nombred'accents propres de notre langue.       Eh bien, q
ui l'e_t cru? Il n'y a pas lé d'endroit oè le b^t blesse pou
r le CPC!"
260 END
270 REM Lignes DATA avec la forme                  des caracter
es francais
280 REM Lettre 'c cedille'
290 DATA 126
300 DATA &x 00000000
310 DATA &x 00000000
```

```
320 DATA &x 00111100
330 DATA &x 01100110
340 DATA &x 01100000
350 DATA &x 01100110
360 DATA &x 00111100
370 DATA &x 00011000
380 Lettre a accent grave
390 DATA 123
400 DATA &x 01100000
410 DATA &x 00110000
420 DATA &x 01111000
430 DATA &x 00001100
440 DATA &x 01111100
450 DATA &x 11001100
460 DATA &x 01110110
470 DATA &x 00000000
480 Lettre o accent circonflexe
490 DATA 124
500 DATA &x 00111100
510 DATA &x 01100110
520 DATA &x 00111100
530 DATA &x 01100110
540 DATA &x 01100110
550 DATA &x 01100110
560 DATA &x 00111100
570 DATA &x 00000000
580 Lettre u accent grave
590 DATA 125
600 DATA &x 00110000
610 DATA &x 00011000
620 DATA &x 01100110
630 DATA &x 01100110
640 DATA &x 01100110
650 DATA &x 01100110
660 DATA &x 00111111
670 DATA &x 00000000
680 Lettre e accent grave
690 DATA 91
700 DATA &x 00110000
710 DATA &x 00011000
```

```
720 DATA &x 00111100
730 DATA &x 01100110
740 DATA &x 01111110
750 DATA &x 01100000
760 DATA &x 00111100
770 DATA &x 00000000
780 Lettre e accent aigu
790 DATA 92
800 DATA &x 00001100
810 DATA &x 00011000
820 DATA &x 00111100
830 DATA &x 01100110
840 DATA &x 01111110
850 DATA &x 01100000
860 DATA &x 00111100
870 DATA &x 00000000
880 Lettre e accent circonflexe
890 DATA 93
900 DATA &x 00111100
910 DATA &x 01100110
920 DATA &x 00111100
930 DATA &x 01100110
940 DATA &x 01111110
950 DATA &x 01100000
960 DATA &x 00111100
970 DATA &x 00000000
980 Lettre a accent circonflexe
990 DATA 94
1000 DATA &x 01111000
1010 DATA &x 11001100
1020 DATA &x 01111000
1030 DATA &x 00001100
1040 DATA &x 01111100
1050 DATA &x 11001100
1060 DATA &x 01110110
1070 DATA &x 00000000
1080 Lettre u accent circonflexe
1090 DATA 95
1100 DATA &x 00111100
1110 DATA &x 01100110
```

```
1120 DATA &x 00000000
1130 DATA &x 01100110
1140 DATA &x 01100110
1150 DATA &x 01100110
1160 DATA &x 00111111
1170 DATA &x 00000000
1180 Lettre i accent circonflexe
1190 DATA 96
1200 DATA &x 00111100
1210 DATA &x 01100110
1220 DATA &x 00011000
1230 DATA &x 00011000
1240 DATA &x 00011000
1250 DATA &x 00011000
1260 DATA &x 00111100
1270 DATA &x 00000000
```

Jeu de signes mathématiques

==========================

Ce programme ne comprend que deux sections:

Les lignes 410 à 2560 contiennent en ligne de DATA 24 signes mathématiques couramment utilisés.
Il s'agit exactement des signes suivants, que vous pouvez appeler en appuyant sur la touche CTRL et la touche figurant en regard de chaque signe:

Alpha	a	Beta	b	Gamma	c
n'appartient pas à	d	appartient à	e	Intégral	f
Pi	g	Sigma	h	Omega	i
infini	j	angle	k	Triangle	l
Carré	n	Cercle	o	parallèle	q
Rectangle	r	Congru	s	Proche de	t
Ensemble vide	u	1 à 1	v	différent de	w
Val. approchée	x	Proportionnel	y	approché à	z

Nous avons choisi de représenter les formes de ces signes en nombres binaires de façon à ce que vous puissiez aisément les modifier si vous n'êtes pas satisfaits des formes figurant en lignes DATA. Mais faites attention à ne fournir que 8 caractères par ligne (des 0 ou des 1: 0 signifie pixel éteint, 1 signifie pixel allumé).

Chaque signe se compose de 8 lignes de 8 chiffres binaires. Si vous ajoutez à un signe quelconque une ligne supplémentaire, le

jeu des caractères suivants sera complètement modifié (vous pouvez fort bien essayer pour voir ce que celà donne!).

Les lignes 80 à 390 lisent les données avec l'instruction READ. Pour que la représentation binaire des différents signes soit bien interprétée comme une suite de valeurs numériques qui puissent être transformées en nombres décimaux pour leur traitement par l'ordinateur, il faut ajouter à chaque chaîne de 0 et de 1 les caractères '&x' qui désignent les nombres binaires.

Après avoir lancé le programme avec 'RUN', vous pouvez afficher à l'écran les signes mathématiques nouvellement créés en appuyant sur la touche CTRL et sur la touche correspondant au signe voulu dans la table de la page précédente.

Pour que le jeu normal de caractères ASCII puisse continuer à être utilisé, nous avons affecté nos signes mathématiques aux codes 1 à 26 qui ne correspondent pas normalement à des caractères affichables. Nous n'avons cependant pas modifié le code ASCII 13 qui correspond à la fonction de la touche ENTER ni le code ASCII 16 qui correspond à la fonction de la touche DEL.

```
10 REM Jeu de caracteres mathematiques
20 REM Programmes Basic pour le CPC464
30 REM Copyright DATA BECKER &
31 '      Rainer Lueers
40 RESTORE
50 SYMBOL AFTER 1
60 REM Reaffectation des
61 '      caracteres ASCII 1 a 12
70 REM pouvant etre appeles
71 '      avec <CTRL> et la touche
80 FOR m=1 TO 12
90 FOR n=1 TO 8
100 READ a$
110 REM Conversion du nom de la chaine
111 '      en nombre binaire puis decimal
120 a$(n)="&x "+a$
130 a(n)=VAL("&x "+a$)
140 NEXT n
150 SYMBOL m,a(1),a(2),a(3),a(4),a(5),a(6),a(7),a(8)
160 NEXT m
170 REM Reaffectation des
171 '      caracteres ASCII 14 a 15
180 '      pouvant etre appeles
181 '      avec <CTRL> et la touche
190 FOR m=14 TO 15
200 FOR n=1 TO 8
210 READ a$
220 REM Conversion du nom de la chaine
222 '      en nombre binaire puis decimal
230 a$(n)="&x "+a$
240 a(n)=VAL("&x "+a$)
250 NEXT n
260 SYMBOL m,a(1),a(2),a(3),a(4),a(5),a(6),a(7),a(8)
270 NEXT m
280 REM Reaffectation des
281 '      caracteres ASCII 17 a 26
290 '      pouvant etre appeles
291 '      avec <CTRL> et une touche
300 FOR m=17 TO 26
310 FOR n=1 TO 8
```

```
320 READ a$
330 REM Conversion du nom de la chaine
331 '       en nombre binaire puis decimal
340 a$(n)="&x "+a$
350 a(n)=VAL("&x "+a$)
360 NEXT n
370 SYMBOL m,a(1),a(2),a(3),a(4),a(5),a(6),a(7),a(8)
380 NEXT m
390 END
400 REM Voici la representation binaire
401 '       du jeu de caracteres mathema-
402 '       tiques.
403 '       Le caractere correspondant a
404 '       chaque groupe de DATA est
405 '       affiche en ligne REM
410 REM Alpha
420 DATA 00000000
430 DATA 00011100
440 DATA 00100100
450 DATA 01000100
460 DATA 01000100
470 DATA 00100100
480 DATA 00011100
490 DATA 00000010
500 REM Beta
510 DATA 00000000
520 DATA 01111000
530 DATA 01000100
540 DATA 01011000
550 DATA 01000100
560 DATA 01000100
570 DATA 01111000
580 DATA 10000000
590 REM Gamma
600 DATA 00000000
610 DATA 10000010
620 DATA 01000100
630 DATA 00101000
640 DATA 00010000
650 DATA 00010000
```

```
660 DATA 00010000
670 DATA 00010000
680 REM n'appartient pas a
690 DATA 00000000
700 DATA 00000100
710 DATA 00011110
720 DATA 00100100
730 DATA 00111110
740 DATA 00100100
750 DATA 00011110
760 DATA 00000100
770 REM appartient a
780 DATA 00000000
790 DATA 00000000
800 DATA 00011110
810 DATA 00100000
820 DATA 00111110
830 DATA 00100000
840 DATA 00011110
850 DATA 00000000
860 REM Entier
870 DATA 00110000
880 DATA 00101000
890 DATA 00101000
900 DATA 00100000
910 DATA 00100000
920 DATA 10100000
930 DATA 10100000
940 DATA 01100000
950 REM Pi
960 DATA 00000000
970 DATA 00000000
980 DATA 01111110
990 DATA 00100100
1000 DATA 00100100
1010 DATA 00100100
1020 DATA 00100100
1030 DATA 00000000
1040 REM Sigma
1050 DATA 00000000
```

```
1060 DATA 01111111
1070 DATA 00100001
1080 DATA 00010000
1090 DATA 00010000
1100 DATA 00100001
1110 DATA 01111111
1120 DATA 00000000
1130 REM Omega
1140 DATA 00000000
1150 DATA 00111100
1160 DATA 00100100
1170 DATA 01000010
1180 DATA 01000010
1190 DATA 00100100
1200 DATA 11100111
1210 DATA 00000000
1220 REM infini
1230 DATA 00000000
1240 DATA 00000000
1250 DATA 01101100
1260 DATA 10010010
1270 DATA 10010010
1280 DATA 10010010
1290 DATA 01101100
1300 DATA 00000000
1310 DATA 00000000
1320 REM angle
1330 DATA 00000010
1340 DATA 00000100
1350 DATA 00001000
1360 DATA 00010000
1370 DATA 00100000
1380 DATA 01000000
1390 DATA 11111110
1400 REM triangle
1410 DATA 00000000
1420 DATA 00000000
1430 DATA 00000000
1440 DATA 00010000
1450 DATA 00101000
```

```
1460 DATA 01000100
1470 DATA 11111110
1480 DATA 00000000
1490 REM carre
1500 DATA 11111110
1510 DATA 10000010
1520 DATA 10000010
1530 DATA 10000010
1540 DATA 10000010
1550 DATA 10000010
1560 DATA 11111110
1570 DATA 00000000
1580 REM cercle
1590 DATA 00000000
1600 DATA 01111100
1610 DATA 10000010
1620 DATA 10000010
1630 DATA 10000010
1640 DATA 10000010
1650 DATA 01111100
1660 DATA 00000000
1670 REM parallele a
1680 DATA 01001000
1690 DATA 01001000
1700 DATA 01001000
1710 DATA 01001000
1720 DATA 01001000
1730 DATA 01001000
1740 DATA 01001000
1750 DATA 01001000
1760 REM angle droit de
1770 DATA 00010000
1780 DATA 00010000
1790 DATA 00010000
1800 DATA 00010000
1810 DATA 00010000
1820 DATA 00010000
1830 DATA 11111110
1840 DATA 00000000
1850 REM congruent
```

```
1860 DATA 01100000
1870 DATA 10011001
1880 DATA 00000110
1890 DATA 00000000
1900 DATA 11111111
1910 DATA 00000000
1920 DATA 11111111
1930 DATA 00000000
1940 REM peu different, proportionnel
1950 DATA 00000000
1960 DATA 00000000
1970 DATA 00000000
1980 DATA 01100000
1990 DATA 10011001
2000 DATA 00000110
2010 DATA 00000000
2020 DATA 00000000
2030 REM ensemble vide
2040 DATA 00000010
2050 DATA 01111100
2060 DATA 10001010
2070 DATA 10010010
2080 DATA 10100010
2090 DATA 01111100
2100 DATA 10000000
2110 DATA 00000000
2120 REM 1 pour 1
2130 DATA 00000000
2140 DATA 00100100
2150 DATA 01000010
2160 DATA 11111111
2170 DATA 01000010
2180 DATA 00100100
2190 DATA 00000000
2200 DATA 00000000
2210 REM different
2220 DATA 00000001
2230 DATA 00000010
2240 DATA 11111111
2250 DATA 00001000
```

```
2260 DATA 11111111
2270 DATA 00100000
2280 DATA 01000000
2290 DATA 00000000
2300 REM approche
2310 DATA 00000000
2320 DATA 01100000
2330 DATA 10011001
2340 DATA 00000110
2350 DATA 01100000
2360 DATA 10011001
2370 DATA 00000110
2380 DATA 00000000
2390 REM proportionnel a
2400 DATA 00000000
2410 DATA 00000000
2420 DATA 01110111
2430 DATA 10001000
2440 DATA 10001000
2450 DATA 10001000
2460 DATA 01110111
2470 DATA 00000000
2480 REM valeur approchee de ...
2490 DATA 00000000
2500 DATA 00000100
2510 DATA 00000010
2520 DATA 11111111
2530 DATA 00000010
2540 DATA 00000100
2550 DATA 00000000
2560 DATA 00000000
```

Jeu de caractères informatiques

============================

Il faut bien reconnaître que le jeu de caractères du CPC est très esthétique. Bien que les différentes lettres ne soient dessinées, comme sur d'autres ordinateurs, que dans une matrice de 8*8 (8 colonnes de 8 points allumés ou éteints), la qualité du dessin des différentes lettres rappelle vraiment la machine à écrire.

Pourquoi vouloir donc créer un nouveau jeu de caractères? Eh bien, celà fait maintenant plusieurs années que dure la grande mode des jeux électroniques. Et pour ces jeux futuristes, il est clair que ce n'est pas de lettres classiques mais bien de lettres et de symboles informatiques qu'on veut pouvoir se servir!

Le programme comprend deux parties principales:

1) Un océan de lignes de DATA comportant le jeu de caractères informatiques
2) La routine de traitement et de chargement des DATA.

Quelques précisions sur la deuxième partie du programme: le CPC présente tout (y compris les lettres et autres caractères) à l'écran sous une forme grafique. Il est donc, contrairement à ce qui est le cas sur les autres ordinateurs, relativement facile, non seulement de mélanger à l'écran texte et graphisme, mais aussi de représenter la même lettre plusieurs fois avec des formes différentes. Nous avons utilisé cette possibilité dans notre programme qui vous permet soit de transformer le jeu de caractères complet dans écriture informatique, soit de travailler simultanément avec les jeux de caractères normaux et

informatiques.

Si vous connaissez la puissante instruction SYMBOL, vous comprendrez certainement comment nous avons pu modifier complètement le jeu de caractères de votre ordinateur.

Mais comment peut-on conserver en mémoire deux jeux de caractères simultanément? Il faut pour celà que le programme continue de tourner car il n'est pas si facile d'appeler simultanément autant de caractères avec les touches du clavier. C'est pourquoi nous avons placé dans le programme un commutateur entre les deux jeux de caractères: si vous appuyez sur TAB vous passez de l'écriture normale à l'écriture informatique ou vice versa.

Voici la technique de programmation que nous avons utilisée: le code ASCII du caractère voulu est interprété (par exemple '!'=33); puis le programme cherche la forme de caractère informatique correspondante et... celle-ci n'est pas placée avec l'instruction SYMBOL à la place de la forme normale du caractère choisi (sinon il ne serait plus possible de revenir au jeu de caractères normal)... mais elle est placée dans un caractère inutilisé (CHR$(255)). C'est donc toujours ce même caractère qui est affiché, chaque fois avec une forme différente, lorsque vous affichez un caractère informatique.

```
10 REM Ecriture informatique
20 REM Programmes Basic pour le CPC464
30 REM Copyright 1984 DATA BECKER &
31 '      Rainer Lueers
40 MODE 1
50 DIM a(122,8)
60 REM liberer la zone de caracteres
61 '      a redefinir
70 REM Lecture de l'ecriture infor-
71 '      matique avec la variable a
80 SYMBOL AFTER 32
90 REM Charger provisoirement le code
91 '      et la forme du caractere dans
92 '      la variable a
100 FOR n=1 TO 73
110 READ a
120 FOR m=1 TO 8
130 READ a$
140 REM Conversion des chaines en
141 '      nombres decimaux
150 a$="&x "+a$
160 a(a,m)=VAL(a$)
170 NEXT m
180 NEXT n
190 REM Options possibles:
191 '      1) entrer un texte en ecriture
192 '       informatique ou normale
193 '       pendant le cours du programme
195 '      2) transformer le jeu de
196 '        caracteres complet en
197 '        ecriture informatique
200 PRINT"Entrez un texte quelconque!             Si vous vo
ulez qu'il apparaisse en       ecriture informatique, appuyez
 d'abord  sur la touche TAB.                      Si vous vo
ulez que tous les caracteres"
210 PRINT"apparaissent maintenant en ecriture      informatiq
ue (y compris le listing),    appuyez maintenant sur la touc
he o";:a$="":INPUT a$
220 a$=UPPER$(a$):IF a$="O" THEN GOTO 510
230 REM Le jeu de caracteres normal est
```

```
231 ' conserve et les lettres ne sont
232 ' converties en ecriture informati-
233 ' que que une par une a votre
234 ' demande
240 PRINT
250 PRINT "Veuillez entrer du texte"
260 PRINT
270 a$=INKEY$
280 IF a$="" THEN GOTO 270
290 REM Petit editeur de texte:
291 '  avec interpretation des touches
292 '  <DELETE>=127
300 IF ASC(a$)=9 THEN GOTO 360
310 IF ASC(a$)=13 THEN PRINT:GOTO 270
320 IF ASC(a$)=127 THEN a$="":IF POS(#0)>1 THEN LOCATE POS(#
0)-1,VPOS(#0):PRINT " ";:LOCATE POS(#0)-1,VPOS(#0):GOTO 270
330 PRINT a$;
340 GOTO 270
350 END
360 a$=INKEY$
370 IF a$="" THEN GOTO 360
380 REM Petit editeur de texte:
381 '  avec interpretation des
382 '  touches <ENTER>=13, <TAB>=9 et
383 '  <DELETE>=127
390 IF ASC(a$)=9 THEN GOTO 270
400 IF ASC(a$)=13 THEN PRINT:GOTO 360
410 IF ASC(a$)=127 THEN a$="":IF POS(#0)>1 THEN LOCATE POS(#
0)-1,VPOS(#0):PRINT " ";:LOCATE POS(#0)-1,VPOS(#0):GOTO 360
ELSE GOTO 360
420 IF ASC(a$)>122 THEN PRINT a$;:GOTO 360
430 IF a(ASC(a$),1)=0 AND a(ASC(a$),2)=0 AND a(ASC(a$),3)=0
AND a(ASC(a$),4)=0 AND a(ASC(a$),5)=0 AND a(ASC(a$),6)=0 AND
 a(ASC(a$),7)=0 AND a(ASC(a$),8)=0 THEN PRINT a$;:GOTO 360
440 REM Pour que le jeu de caracteres
441 '  normal ne soit pas detruit,
442 '  chaque caractere est represente
443 '  par transformation de CHR$(255)
450 z=ASC(a$)
460 SYMBOL 255,a(z,1),a(z,2),a(z,3),a(z,4),a(z,5),a(z,6),a(z
```

```
,7),a(z,8)
470 PRINT CHR$(255);
480 GOTO 360
490 END
500 REM Lecture de l'ecriture
501 '    informatique et modification
502 '    directe du jeu de caracteres
503 '    normal
510 RESTORE
520 FOR n=1 TO 73
530 READ m
540 FOR o=1 TO 8
550 READ p$
560 REM Conversion des chaines en
561 '   nombres binaires puis en
562 '   nombres decimaux
570 p$="&x "+p$
580 b(o)=VAL(p$)
590 NEXT o
600 SYMBOL m,b(1),b(2),b(3),b(4),b(5),b(6),b(7),b(8)
610 NEXT n
620 END
630 REM !
640 DATA 33
650 DATA 00111000
660 DATA 00111000
670 DATA 00111000
680 DATA 00111000
690 DATA 00011000
700 DATA 00000000
710 DATA 00011000
720 DATA 00000000
730 REM "
740 DATA 34
750 DATA 01100110
760 DATA 01100110
770 DATA 01100110
780 DATA 00000000
790 DATA 00000000
800 DATA 00000000
```

```
810 DATA 00000000
820 DATA 00000000
830 REM #
840 DATA 35
850 DATA 01100110
860 DATA 11111111
870 DATA 01100110
880 DATA 01100110
890 DATA 11111111
900 DATA 01100110
910 DATA 00000000
920 DATA 00000000
930  REM '
940  DATA 39
950  DATA 00011000
960  DATA 00011000
970  DATA 00011000
980  DATA 00000000
990  DATA 00000000
1000 DATA 00000000
1010 DATA 00000000
1020 DATA 00000000
1030 REM (
1040 DATA 40
1050 DATA 00011110
1060 DATA 00011000
1070 DATA 00011000
1080 DATA 00111000
1090 DATA 00111000
1100 DATA 00111000
1110 DATA 00111110
1120 DATA 00000000
1130 REM )
1140 DATA 41
1150 DATA 01111000
1160 DATA 00011000
1170 DATA 00011000
1180 DATA 00011100
1190 DATA 00011100
1200 DATA 00011100
```

```
1210 DATA 01111100
1220 DATA 00000000
1230 REM 0
1240 DATA 48
1250 DATA 01111111
1260 DATA 01100011
1270 DATA 01100011
1280 DATA 01100011
1290 DATA 01100011
1300 DATA 01100011
1310 DATA 01111111
1320 DATA 00000000
1330 REM 1
1340 DATA 49
1350 DATA 00111000
1360 DATA 00011000
1370 DATA 00011000
1380 DATA 00011000
1390 DATA 00111110
1400 DATA 00111110
1410 DATA 00111110
1420 DATA 00000000
1430 REM 2
1440 DATA 50
1450 DATA 01111111
1460 DATA 00000011
1470 DATA 00000011
1480 DATA 01111111
1490 DATA 01100000
1500 DATA 01100000
1510 DATA 01111111
1520 DATA 00000000
1530 REM 3
1540 DATA 51
1550 DATA 01111110
1560 DATA 00000110
1570 DATA 00000110
1580 DATA 01111111
1590 DATA 00000111
1600 DATA 00000111
```

```
1610 DATA 11111111
1620 DATA 00000000
1630 REM 4
1640 DATA 52
1650 DATA 01110000
1660 DATA 01110000
1670 DATA 01110000
1680 DATA 01110111
1690 DATA 01110111
1700 DATA 01111111
1710 DATA 00000111
1720 DATA 00000000
1730 REM 5
1740 DATA 53
1750 DATA 01111111
1760 DATA 01100000
1770 DATA 01100000
1780 DATA 01111111
1790 DATA 00000111
1800 DATA 00000111
1810 DATA 01111111
1820 DATA 00000000
1830 REM 6
1840 DATA 54
1850 DATA 01111100
1860 DATA 01101100
1870 DATA 01100000
1880 DATA 01111111
1890 DATA 01100011
1900 DATA 01100011
1910 DATA 01111111
1920 DATA 00000000
1930 REM 7
1940 DATA 55
1950 DATA 01111111
1960 DATA 00000011
1970 DATA 00000011
1980 DATA 00011111
1990 DATA 00011000
2000 DATA 00011000
```

```
2010 DATA 00011000
2020 DATA 00000000
2030 REM 8
2040 DATA 56
2050 DATA 00111110
2060 DATA 00110110
2070 DATA 00110110
2080 DATA 01111111
2090 DATA 01110111
2100 DATA 01110111
2110 DATA 01111111
2120 DATA 00000000
2130 REM 9
2140 DATA 57
2150 DATA 01111111
2160 DATA 01100011
2170 DATA 01100011
2180 DATA 01111111
2190 DATA 00000111
2200 DATA 00000111
2210 DATA 00000111
2220 DATA 00000000
2230 REM :
2240 DATA 58
2250 DATA 00000000
2260 DATA 00011000
2270 DATA 00011000
2280 DATA 00000000
2290 DATA 00011000
2300 DATA 00011000
2310 DATA 00000000
2320 DATA 00000000
2330 REM ;
2340 DATA 59
2350 DATA 00000000
2360 DATA 00011000
2370 DATA 00011000
2380 DATA 00000000
2390 DATA 00011000
2400 DATA 00011000
```

```
2410 DATA 00110000
2420 DATA 00000000
2430 REM =
2440 DATA 61
2450 DATA 00000000
2460 DATA 01111110
2470 DATA 00000000
2480 DATA 00000000
2490 DATA 01111110
2500 DATA 00000000
2510 DATA 00000000
2520 DATA 00000000
2530 REM ?
2540 DATA 63
2550 DATA 01111111
2560 DATA 01100011
2570 DATA 00000011
2580 DATA 00011111
2590 DATA 00011100
2600 DATA 00000000
2610 DATA 00011100
2620 DATA 00000000
2630 REM à
2640 DATA 64
2650 DATA 01111111
2660 DATA 01100011
2670 DATA 01101111
2680 DATA 01101111
2690 DATA 01101111
2700 DATA 01100000
2710 DATA 01111111
2720 DATA 00000000
2730 REM A
2740 DATA 65
2750 DATA 00111111
2760 DATA 00110011
2770 DATA 00110011
2780 DATA 01111111
2790 DATA 01110011
2800 DATA 01110011
```

```
2810 DATA 01110011
2820 DATA 00000000
2830 REM B
2840 DATA 66
2850 DATA 01111110
2860 DATA 01100110
2870 DATA 01100110
2880 DATA 01111111
2890 DATA 01100111
2900 DATA 01100111
2910 DATA 01111111
2920 DATA 00000000
2930 REM C
2940 DATA 67
2950 DATA 01111111
2960 DATA 01100111
2970 DATA 01100111
2980 DATA 01100000
2990 DATA 01100011
3000 DATA 01100011
3010 DATA 01111111
3020 DATA 00000000
3030 REM D
3040 DATA 68
3050 DATA 01111110
3060 DATA 01100110
3070 DATA 01100110
3080 DATA 01110111
3090 DATA 01110111
3100 DATA 01110111
3110 DATA 01111111
3120 DATA 00000000
3130 REM E
3140 DATA 69
3150 DATA 01111111
3160 DATA 01100000
3170 DATA 01100000
3180 DATA 01111111
3190 DATA 01110000
3200 DATA 01110000
```

```
3210 DATA 01111111
3220 DATA 00000000
3230 REM F
3240 DATA 70
3250 DATA 01111111
3260 DATA 01100000
3270 DATA 01100000
3280 DATA 01111111
3290 DATA 01110000
3300 DATA 01110000
3310 DATA 01110000
3320 DATA 00000000
3330 REM G
3340 DATA 71
3350 DATA 01111111
3360 DATA 01100011
3370 DATA 01100000
3380 DATA 01101111
3390 DATA 01100111
3400 DATA 01100111
3410 DATA 01111111
3420 DATA 00000000
3430 REM H
3440 DATA 72
3450 DATA 01110011
3460 DATA 01110011
3470 DATA 01110011
3480 DATA 01111111
3490 DATA 01110011
3500 DATA 01110011
3510 DATA 01110011
3520 DATA 00000000
3530 REM I
3540 DATA 73
3550 DATA 00001100
3560 DATA 00001100
3570 DATA 00001100
3580 DATA 00001100
3590 DATA 00111100
3600 DATA 00111100
```

```
3610 DATA 00111100
3620 DATA 00000000
3630 REM J
3640 DATA 74
3650 DATA 00001100
3660 DATA 00001100
3670 DATA 00001100
3680 DATA 00001110
3690 DATA 00001110
3700 DATA 01101110
3710 DATA 01111110
3720 DATA 00000000
3730 REM K
3740 DATA 75
3750 DATA 01100110
3760 DATA 01100110
3770 DATA 01101100
3780 DATA 01111111
3790 DATA 01100111
3800 DATA 01100111
3810 DATA 01100111
3820 DATA 00000000
3830 REM L
3840 DATA 76
3850 DATA 00110000
3860 DATA 00110000
3870 DATA 00110000
3880 DATA 01110000
3890 DATA 01110000
3900 DATA 01110000
3910 DATA 01111110
3920 DATA 00000000
3930 REM M
3940 DATA 77
3950 DATA 01100111
3960 DATA 01111111
3970 DATA 01111111
3980 DATA 01110111
3990 DATA 01100111
4000 DATA 01100111
```

```
4010 DATA 01100111
4020 DATA 00000000
4030 REM N
4040 DATA 78
4050 DATA 01100111
4060 DATA 01110111
4070 DATA 01111111
4080 DATA 01101111
4090 DATA 01100111
4100 DATA 01100111
4110 DATA 01100111
4120 DATA 00000000
4130 REM O
4140 DATA 79
4150 DATA 01111111
4160 DATA 01100011
4170 DATA 01100011
4180 DATA 01100111
4190 DATA 01100111
4200 DATA 01100111
4210 DATA 01111111
4220 DATA 00000000
4230 REM P
4240 DATA 80
4250 DATA 01111111
4260 DATA 01100011
4270 DATA 01100011
4280 DATA 01111111
4290 DATA 01110000
4300 DATA 01110000
4310 DATA 01110000
4320 DATA 00000000
4330 REM Q
4340 DATA 81
4350 DATA 01111111
4360 DATA 01100011
4370 DATA 01100011
4380 DATA 01100111
4390 DATA 01100111
4400 DATA 01100111
```

```
4410 DATA 01111111
4420 DATA 00000111
4430 REM R
4440 DATA 82
4450 DATA 01111110
4460 DATA 01100110
4470 DATA 01100110
4480 DATA 01111111
4490 DATA 01110111
4500 DATA 01110111
4510 DATA 01110111
4520 DATA 00000000
4530 REM S
4540 DATA 83
4550 DATA 01111111
4560 DATA 01100000
4570 DATA 01111111
4580 DATA 00000011
4590 DATA 01110011
4600 DATA 01110011
4610 DATA 01111111
4620 DATA 00000000
4630 REM T
4640 DATA 84
4650 DATA 01111111
4660 DATA 00011100
4670 DATA 00011100
4680 DATA 00011100
4690 DATA 00011100
4700 DATA 00011100
4710 DATA 00011100
4720 DATA 00000000
4730 REM U
4740 DATA 85
4750 DATA 01100111
4760 DATA 01100111
4770 DATA 01100111
4780 DATA 01100111
4790 DATA 01100111
4800 DATA 01100111
```

```
4810 DATA 01111111
4820 DATA 00000000
4830 REM V
4840 DATA 86
4850 DATA 01100111
4860 DATA 01100111
4870 DATA 01100111
4880 DATA 01100111
4890 DATA 01101111
4900 DATA 00111110
4910 DATA 00011100
4920 DATA 00000000
4930 REM W
4940 DATA 87
4950 DATA 01100111
4960 DATA 01100111
4970 DATA 01100111
4980 DATA 01101111
4990 DATA 01111111
5000 DATA 01111111
5010 DATA 01100111
5020 DATA 00000000
5030 REM X
5040 DATA 88
5050 DATA 01110011
5060 DATA 01110011
5070 DATA 01110011
5080 DATA 00111110
5090 DATA 01100111
5100 DATA 01100111
5110 DATA 01100111
5120 DATA 00000000
5130 REM Y
5140 DATA 89
5150 DATA 01100111
5160 DATA 01100111
5170 DATA 01100111
5180 DATA 01111111
5190 DATA 00011100
5200 DATA 00011100
```

```
5210 DATA 00011100
5220 DATA 00000000
5230 REM Z
5240 DATA 90
5250 DATA 01111111
5260 DATA 01100110
5270 DATA 01101100
5280 DATA 00011000
5290 DATA 00110111
5300 DATA 01100111
5310 DATA 01111111
5320 DATA 00000000
5330 REM a
5340 DATA 97
5350 DATA 00000000
5360 DATA 00000000
5370 DATA 00111110
5380 DATA 00000110
5390 DATA 01111110
5400 DATA 01100110
5410 DATA 01111110
5420 DATA 00000000
5430 REM b
5440 DATA 98
5450 DATA 00000000
5460 DATA 01110000
5470 DATA 01110000
5480 DATA 01111110
5490 DATA 01110110
5500 DATA 01110110
5510 DATA 01111110
5520 DATA 00000000
5530 REM c
5540 DATA 99
5550 DATA 00000000
5560 DATA 00000000
5570 DATA 01111100
5580 DATA 01110000
5590 DATA 01110000
5600 DATA 01110000
```

```
5610 DATA 01111100
5620 DATA 00000000
5630 REM d
5640 DATA 100
5650 DATA 00000000
5660 DATA 00000110
5670 DATA 00000110
5680 DATA 01111110
5690 DATA 01101110
5700 DATA 01101110
5710 DATA 01111110
5720 DATA 00000000
5730 REM e
5740 DATA 101
5750 DATA 00000000
5760 DATA 00000000
5770 DATA 01111110
5780 DATA 01100110
5790 DATA 01111110
5800 DATA 01100000
5810 DATA 01111110
5820 DATA 00000000
5830 REM f
5840 DATA 102
5850 DATA 00000000
5860 DATA 00011110
5870 DATA 00011000
5880 DATA 00111110
5890 DATA 00011000
5900 DATA 00011000
5910 DATA 00011000
5920 DATA 00000000
5930 REM g
5940 DATA 103
5950 DATA 00000000
5960 DATA 00000000
5970 DATA 01111110
5980 DATA 01101110
5990 DATA 01101110
6000 DATA 01111110
```

```
6010 DATA 00001110
6020 DATA 01111110
6030 REM h
6040 DATA 104
6050 DATA 00000000
6060 DATA 01110000
6070 DATA 01110000
6080 DATA 01111100
6090 DATA 01110110
6100 DATA 01110110
6110 DATA 01110110
6120 DATA 00000000
6130 REM i
6140 DATA 105
6150 DATA 00000000
6160 DATA 00011000
6170 DATA 00000000
6180 DATA 00011000
6190 DATA 00011000
6200 DATA 00111100
6210 DATA 00111100
6220 DATA 00000000
6230 REM j
6240 DATA 106
6250 DATA 00000000
6260 DATA 00001110
6270 DATA 00000000
6280 DATA 00001110
6290 DATA 00001110
6300 DATA 00001110
6310 DATA 00001110
6320 DATA 00111110
6330 REM k
6340 DATA 107
6350 DATA 00000000
6360 DATA 01100000
6370 DATA 01100000
6380 DATA 01101100
6390 DATA 01111000
6400 DATA 01101100
```

```
6410 DATA 01100110
6420 DATA 00000000
6430 REM l
6440 DATA 108
6450 DATA 00000000
6460 DATA 00111100
6470 DATA 00011100
6480 DATA 00011100
6490 DATA 00011100
6500 DATA 00011100
6510 DATA 00011100
6520 DATA 00000000
6530 REM m
6540 DATA 109
6550 DATA 00000000
6560 DATA 00000000
6570 DATA 01100111
6580 DATA 01111111
6590 DATA 01111111
6600 DATA 01101011
6610 DATA 01100011
6620 DATA 00000000
6630 REM n
6640 DATA 110
6650 DATA 00000000
6660 DATA 00000000
6670 DATA 01111110
6680 DATA 01111110
6690 DATA 01100110
6700 DATA 01100110
6710 DATA 01100110
6720 DATA 00000000
6730 REM o
6740 DATA 111
6750 DATA 00000000
6760 DATA 00000000
6770 DATA 01111110
6780 DATA 01101110
6790 DATA 01101110
6800 DATA 01101110
```

```
6810 DATA 01111110
6820 DATA 00000000
6830 REM p
6840 DATA 112
6850 DATA 00000000
6860 DATA 00000000
6870 DATA 01111110
6880 DATA 01110110
6890 DATA 01110110
6900 DATA 01111110
6910 DATA 01110000
6920 DATA 01110000
6930 REM q
6940 DATA 113
6950 DATA 00000000
6960 DATA 00000000
6970 DATA 01111110
6980 DATA 01101110
6990 DATA 01101110
7000 DATA 01111110
7010 DATA 00001110
7020 DATA 00001110
7030 REM r
7040 DATA 114
7050 DATA 00000000
7060 DATA 00000000
7070 DATA 01111110
7080 DATA 01110110
7090 DATA 01110000
7100 DATA 01110000
7110 DATA 01110000
7120 DATA 00000000
7130 REM s
7140 DATA 115
7150 DATA 00000000
7160 DATA 00000000
7170 DATA 01111110
7180 DATA 01100000
7190 DATA 01111110
7200 DATA 00001110
```

```
7210 DATA 01111110
7220 DATA 00000000
7230 REM t
7240 DATA 116
7250 DATA 00000000
7260 DATA 00111000
7270 DATA 01111110
7280 DATA 00111000
7290 DATA 00111000
7300 DATA 00111000
7310 DATA 00111110
7320 DATA 00000000
7330 REM u
7340 DATA 117
7350 DATA 00000000
7360 DATA 00000000
7370 DATA 01101110
7380 DATA 01101110
7390 DATA 01101110
7400 DATA 01101110
7410 DATA 01111110
7420 DATA 00000000
7430 REM v
7440 DATA 118
7450 DATA 00000000
7460 DATA 00000000
7470 DATA 01101110
7480 DATA 01101110
7490 DATA 01101110
7500 DATA 00111100
7510 DATA 00011000
7520 DATA 00000000
7530 REM w
7540 DATA 119
7550 DATA 00000000
7560 DATA 00000000
7570 DATA 01100011
7580 DATA 01101011
7590 DATA 01111111
7600 DATA 01111111
```

```
7610 DATA 00110110
7620 DATA 00000000
7630 REM x
7640 DATA 120
7650 DATA 00000000
7660 DATA 00000000
7670 DATA 01100110
7680 DATA 00111100
7690 DATA 00011000
7700 DATA 00111100
7710 DATA 01100110
7720 DATA 00000000
7730 REM y
7740 DATA 121
7750 DATA 00000000
7760 DATA 00000000
7770 DATA 01101110
7780 DATA 01101110
7790 DATA 01101110
7800 DATA 01111110
7810 DATA 00001110
7820 DATA 01111110
7830 REM z
7840 DATA 122
7850 DATA 00000000
7860 DATA 00000000
7870 DATA 01111110
7880 DATA 00001100
7890 DATA 00011000
7900 DATA 00110000
7910 DATA 01111110
7920 DATA 00000000
7930 a$=INKEY$:IF a$="" THEN GOTO 7930
7940 PRINT ASC(a$):GOTO 7930
```

Messages d'erreur

On a parfois l'impression d'être victime d'une malédiction quand, après avoir longuement travaillé au développement d'un programme, on l'entre dans l'ordinateur et qu'on s'aperçoit que malgré des efforts répétés il se produit toujours des erreurs.

Comme le CPC n'a pas été conçu en France, les messages d'erreur sont naturellement affichés en anglais et ils sont de plus relativement brefs pour ne pas consommer trop de place en mémoire (syntax error, type mismatch...).

Les programmeurs consciencieux apprennent tous avec le temps à connaître par coeur la signification des différents messages d'erreur, mais c'est le plus souvent un long chemin semé d'embûches et de petites fautes d'interprétation qui ralentissent le développement des programmes. Comme le CPC dispose de l'instruction ON ERROR GOTO, nous pouvons le forcer à réagir à une erreur comme nous le souhaitons et il peut même corriger certaines erreurs de lui-même: si par exemple le CPC charge des données à partir de la cassette et qu'après avoir lu toutes les données d'un fichier vous tentez de lire d'autres données, il va normalement afficher le message d'erreur: 'OF' met (= rencontré la fin du fichier). Il est alors très simple de prévoir dans une routine de traitement des erreurs que le CPC devra fermer le fichier ouvert quand se produira cette erreur. Le programme pourra ensuite suivre son cours normal.

Notre programme de messages d'erreur ne supprime toutefois pas les

erreurs qui se produisent (ceci peut mener en effet à des résultats inattendus, si la suppression des erreurs n'est pas maniée avec précaution) mais il vous aide très efficacement dans la recherche des erreurs. Le mieux est que vous essayiez vous-même! Entrez en ligne 20 uniquement NEXT n. Si vous lancez le programme, vous voyez que l'erreur est immédiatement détectée et indiquée et qu'une stratégie de correction du programme vous est proposée.

Comment procéder maintenant avec les programmes que vous avez déjà écrits? Il faut d'abord renuméroter les lignes de vos programmes avec l'instruction RENUM pour que les deux programmes ne se chevauchent pas (les numéros de lignes de vos programmes doivent donc être compris entre 10 et 10000). Utilisez maintenant l'instruction 'MERGE' (reportez-vous aux explications du manuel) pour fusionner les deux programmes entre eux. Vous pouvez alors lancer votre programme et le corriger à l'aide des messages d'erreur et des propositions de rectification qui vous sont faites.

Le CPC dispose d'autres instructions que ON ERROR GOTO qui peuvent vous aider dans le domaine du dépistage des erreurs: l'interrogation de la variable 'ERL' vous fournit en effet la ligne où s'est produite l'erreur et la variable 'ERR' vous fournit le numéro du message d'erreur. Vous pouvez enfin produire des erreurs avec l'instruction ERROR.

```
2 PRINT
10 REM Messages d'erreur
11 REM Programmes Basic pour le CPC464
12 REM Copyright 1984 DATA BECKER &
13 '      Rainer Lueers
14 INK 0,1:INK 1,24:INK 3,1,24
15 ON ERROR GOTO 10000
20 REM Veuillez taper votre programme
21 '      entre les lignes 20 et 9999
22 '      ou bien utilisez MERGE
100 STOP
9999 STOP
10000 REM Debut de la routine de
10001 '      traitement detaille des
10002 '      erreurs
10010 MODE 1:PEN 1
10020 PRINT"Erreur en ligne"ERL"!"
10030 GOSUB 11420
10040 REM Unexpected NEXT
10050 IF ERR=1 THEN PRINT "En ligne"ERL"figure l'instruction
 ":PEN 3 ELSE GOTO 10100
10060 PRINT "'NEXT'";:PEN 1:PRINT "sans qu'une instruction '
FOR'             correspondante ne figure auparavant dansle pr
ogramme."
10070 PRINT"Il se peut egalement que vous ayez      utilise
avec NEXT le nom d'une variable correspondant a une autre bo
ucle."
10080 END
10090 REM Syntax Error
10100 IF ERR=2 THEN PRINT "La construction de la ligne"ERL E
LSE GOTO 10150
10110 PRINT "est incorrecte ou un mot a ete mal      ecrit.
On parle de     ":PEN 3:PRINT "SYNTAX ERROR";:PEN 1:PRINT ".
"
10120 PRINT
10130 END
10140 REM Unexpected RETURN
10150 IF ERR=3 THEN PRINT "En ligne"ERL"figure l'instruction
":PEN 3:PRINT "RETURN";:PEN 1:PRINT ", sans qu'une instructi
on "; ELSE GOTO 10190
```

```
10160 PRINT "'GOSUB'":PRINT"n'ait auparavant appele un
        sous-programme correspondant."
10170 END
10180 REM DATA exhausted
10190 IF ERR=4 THEN PRINT "Vous voulez faire lire encore d'a
utres":PEN 3:PRINT"DATAS";:PEN 1:ELSE GOTO 10230
10200 PRINT"Alors que toutes les donnees en DATAS   ont deja
 ete lues. Il se peut egalement que vous ayez oublie d'utili
ser l'ins-  truction RESTORE avant de faire lire    a nouvea
u les donnees en DATAS."
10210 END
10220 REM Improper argument
10230 IF ERR=5 THEN PRINT "Un parametre ou un ";:PEN 3:PRINT
 "argument";:PEN 1:PRINT " a ete" ELSE GOTO 10270
10240 PRINT"entre de maniere incorrecte. En effet,  vous ne
pouvez pas utiliser n'importent quelles valeurs avec les ins
tructions   SOUND ou ENV par exemple. Consultez      votre ma
nuel!"
10250 END
10260 REM Overflow
10270 IF ERR=6 THEN PRINT "Votre ordinateur detecte un ";:PE
N 3:PRINT "OVERFLOW":PEN 1:PRINT"c'est-a-dire un depassement
." ELSE GOTO 10320
10280 PRINT"En effet, vous avez essaye en            ligne"ER
L;:PRINT "de representer un":PRINT"nombre qui depasse les po
ssibilites de  representation du CPC464 (nombres >     1.7E-
38)."
10290 PRINT "Vous avez peut-etre aussi essaye de     convert
ir un nombre en un autre type de nombres, sans respecter la
valeur        limite."
10300 END
10310 REM Memory full
10320 IF ERR=7 THEN PRINT "Memoire ou en anglais ";:PEN 3:PR
INT "MEMORY";:PEN 1:PRINT " pleine." ELSE GOTO 10380
10330 PRINT"Il faut le faire! Vous avez ecrit un    programm
e Basic tellement long qu'il    depasse la capacite de memoi
re de        l'ordinateur. Essayez de trouver ou vous";
10340 PRINT"pourriez raccourcir votre programme,    par exem
ple en supprimant des lignes de REM. Vous pouvez d'autre par
t arriver   tres rapidement a une surcharge de la   memoire
```

```
si vous dimensionnez vos        variables ou tableaux trop l
argement ou ";
10350 PRINT"si vous imbriquez trop de boucles FOR...NEXT ou
trop de sous-programmes."
10360 END
10370 REM Line does not exist
10380 IF ERR=8 THEN PRINT "Vous voulez sauter a une ligne de
       programme qui n'existe pas, en anglais:";:PEN 3:PRINT
 "LINE DOES NOT EXIST";:PEN 1:PRINT "." ELSE GOTO 10420
10390 PRINT"Vous avez peut-etre mis la charrue avantles boeu
fs ou bien vous avez deja       renumerote le programme sans
 tenir      compte de ce meme message d'erreur."
10400 END
10410 REM Subscript out of range
10420 IF ERR=9 THEN PRINT "Le ";:PEN 3:PRINT "SUBSCRIPT";:PE
N 1:PRINT "(la grandeur de la dimension choisie    d'une var
iable) est excessive "; ELSE GOTO 10470
10430 PRINT"en ligne":PRINT ERL:PRINT"Vous avez oublie de di
mensionner une    variable ou vous ne l'avez pas dimen-   si
onnee assez grande."
10450 END
10460 REM Array already dimensioned
10470 IF ERR=10 THEN PRINT "Le dimensionnement avec ";:PEN 3
:PRINT "DIM";:PEN 1:PRINT "a deja ete fait quelque part dans
 le    programme."; ELSE GOTO 10510
10480 PRINT"Vous ne pouvez dimensionner a nouveau   en ligne
";ERL;"sans avoir auparavant":PRINT"utilise l'instruction CL
EAR."
10490 END
10500 REM Division by zero
10510 IF ERR=11 THEN PRINT"La ";:PEN 3:PRINT "DIVISION";:PEN
 1:PRINT " par 0 n'est pas permise.":PRINT "Cherchez donc co
mment vous avez pu en   provoquer une (Verifiez egalement le
s   variables dans la ligne fautive)." ELSE GOTO 10540
10520 END
10530 REM Invalid direct command
10540 IF ER=12 THEN PRINT "Vous ne pouvez entrer cette instr
uction directement: ce n'est pas une ";:PEN 3:PRINT "direct"
:PRINT "command";:PEN 1:PRINT "." ELSE GOTO 10580
10550 PRINT "(par exemple 'A$=INKEY$' pour interrogerle clav
```

```
ier). Integrez cette instruction dans un programme et lancez
-le avec     'RUN'."
10560 END
10570 REM Type mismatch
10580 IF ERR=13 THEN PRINT "Vous avez choisi un mauvais type
 de     variable:";:PRINT "(";:PEN 3:PRINT "TYPE MISMATCH";:
PEN 1:PRINT ").":PRINT "Les nombres ne peuvent etre lus qu'a
vec des variables numeriques (Variables     A,B,C...)"; ELSE
 GOTO 10630
10590 PRINT "Si vous voulez lire (avec READ ou INPUT)une var
iable numerique OU une chaine de caracteres, vous devez empl
oyer une     variable alphanumerique (A$,B$,C$...).";
10600 PRINT"Si vous faites lire un nombre par une   variable
 alphanumerique, vous pouvez    convertir a nouveau la varia
ble-chaine  en nombre, avec l'instruction VAL(...)."
10610 END
10620 REM String space full
10630 IF ERR=14 THEN PRINT"Il n'y a plus de place pour de no
uvelleschaines de caracteres (";:PEN 3:PRINT "STRINGS";:PEN
1:PRINT ").":PRINT"Pour continuer d'entrer des chaines,    v
ous pouvez:" ELSE GOTO 10670
10640 PRINT:PRINT"1) Raccourcir le programme Basic         2)
 Sauvegarder les chaines existantes      sur cassette et eff
acer avec CLEAR la   zone-memoire des chaines.";:PRINT"   Vo
us pourrez ensuite entrer de          nouvelles variables."
10650 END
10660 REM String too long
10670 IF ERR=15 THEN PRINT "La chaine de caracteres (";:PEN
3:PRINT "STRING";:PEN 1:PRINT ")":PRINT"utilisee en ligne";E
RL;"a plus de" ELSE GOTO 10710
10680 PRINT "255 caracteres, ce qui n'est pas permis!Faites
donc attention lorsque vous      agrandissez des chaines ave
c + ou avec  les fonctions de manipulation de chainestelles
que INSTR."
10690 END
10700 REM String expression too complex
10710 IF ERR=16 THEN PRINT "La chaine de caracteres en ligne
 ";ERL:PRINT "est ";:PEN 3:PRINT "trop complexe";:PEN 1:PRIN
T ".":PRINT"Ne compliquez pas artificiellement le   calcul d
'une  chaine en utilisant trop" ELSE GOTO 10750
```

```
10720 PRINT "d'instructions de manipulation de        chaines
 telles que 'LEFT$', 'RIGHT$',    'MID$'ou 'INSTR'.
10730 END
10740 REM Cannot CONTinue
10750 IF ER=17 THEN:PEN 3:PRINT "CONT";:PEN 1:PRINT "inue es
t une fonction interessante" ELSE GOTO 10800
10760 PRINT "puisqu'elle permet presque toujours une reprise
 du programme apres 'END', 'STOP'ou apres une interruption d
u programme  par l'utilisation de la touche <ESC>,    mais  A
TTENTION!"
10770 PRINT"Dans le cas present, il s'est passe tropd'evenem
ents pour que l'ordinateur       puisse encore se souvenir a
partir de   quelle ligne il doit reprendre            l'execut
ion du programme."
10780 END
10790 REM Unknown user function
10800 IF ERR=18 THEN PEN 3:PRINT "FN";:PEN 1:PRINT " est cer
tainement une instruction" ELSE GOTO 10860
10810 PRINT"pratique pour calculer des fonctions    mathemat
iques qui ont ete auparavant    definies dans le programme a
vec DEF FN, mais en l'occurence, vous avez appele    avec FN
une fonction en ligne";ERL;
10820 PRINT"qui n'avait pas encore ete definie.":PRINT:PRINT
"Nous vous conseillons donc de toujours  regrouper les defin
itions en debut de"
10830 PRINT"programme, dans ce qu'on appelle la     partie-d
eclaration du programme.
10840 END
10850 REM RESUME missing
10860 IF ERR=19 THEN PEN 3:PRINT "RESUME";:PEN 1:PRINT " doi
t etre utilise apres l'ins-" ELSE GOTO 10910
10870 PRINT"truction ON ERROR GOTO pour que le      programm
e puisse reprendre son cours au bon endroit:":PRINT"Si vous
entrez simplement RESUME,        l'execution du programme rep
rendra a"
10880 PRINT"partir de l'endroit ou l'erreur est     apparue.
":PRINT"Si vous entrez RESUME plus un numero de ligne, l'exe
cution du programme se      poursuivra a partir de cette lig
ne."
10890 END
```

```
10900 REM Unexpected RESUME
10910 IF ERR=20 THEN PRINT "L'instruction ";:PEN 3:PRINT "RE
SUME":PEN 1:PRINT "est apparue en ligne"ERL ELSE GOTO 10960
10920 PRINT"sans qu'une routine de traitement des   erreurs
n'ait ete auparavant appelee    dans le programme avec ON ER
ROR GOTO.   Il se peut que vous ayez simplement     oublie d
e terminer le programme avec     'END'.";
10930 PRINT" C'est pourquoi le programme a":PRINT"continue e
t execute la routine de        traitement des erreurs."
10940 END
10950 REM Direct command
10960 IF ERR=21 THEN PRINT "Lors du chargement d'un programm
e sur   cassette, une instruction":PRINT"(";:PEN 3:PRINT "DI
RECT COMMAND";:PEN 1:PRINT "),":PRINT"a ete trouvee, qui app
arait dans le" ELSE GOTO 11000
10970 PRINT"programme sans numero de ligne. Ceci    n'est pa
s permis."
10980 END
10990 REM Operand missing
11000 IF ERR=22 THEN PRINT "Certaines instructions BASIC ne
peuvent etre employees sans etre suivies d'un   minimum d'";
:PEN 3:PRINT "OPERANDES";:PEN 1:PRINT ".":PRINT"Par exemple
l'instruction SOUND doit" ELSE GOTO 11030
11010 PRINT"etre suivie au moins de l'indication du numero d
e canal et de la frequence:     SOUND 1,100 par exemple, mai
s pas SOUND."
11020 REM Line too long
11030 IF ERR=23 THEN PRINT "Ligne de programme trop longue":
PRINT"(";:PEN 3:PRINT "too long";:PEN 1:PRINT ")." ELSE GOTO
 11070
11040 PRINT"Une ligne de programme ne peut avoir    plus de
255 caracteres. Vous pouvez     corriger cette erreur par ex
emple en    scindant cette ligne en plusieurs       petites
lignes."
11050 END
11060 REM EOF met
11070 IF ERR=24 THEN PEN 3:PRINT "EOF";:PEN 1:PRINT " signif
ie END OF FILE." ELSE GOTO 11150
11080 PRINT "Vous avez essaye par erreur en ligne":PRINT ERL
;"de charger plus de donnees":PRINT"d'un fichier qu'il n'en
```

```
contient.        Nous pouvons vous donner les conseils   suiv
ants pour eviter cette erreur:"
11090 PRINT"Avant la lecture des donnees sur         cassette
, utilisez le test:                IF EOF THEN CLOSE IN:GOTO ..
."
11100 PRINT"Lorsque EOF sera alors rencontre par    l'ordina
teur, le fichier sera ferme et  le programme se poursuivra e
n un autre  endroit."
11110 PRINT"Autre possibilite: placer le nombre     d'elemen
ts stockes dans un fichier dans le fichier lui-meme. Lors du
 chargement,faites d'abord lire ce nombre, et"
11120 PRINT"utilisez-le dans une boucle FOR ... NEXT."
11130 END
11140 REM File type error
11150 IF ERR=25 THEN PRINT "Le type du fichier (";:PEN 3:PRI
NT "FILE TYPE";:PEN 1:PRINT ")":PRINT"est incorrect." ELSE G
OTO 11200
11160 PRINT"Vous ne pouvez lire des donnees avec    l'instru
ction OPENIN que sur des         fichiers ASCII. Par contre,
LOAD, RUN ouMERGE ne permettent de charger que des  programm
es qui ont ete sauvegardes       directement avec SAVE."
11170 PRINT"Ces programmes n'ont pas ete en effet   sauvegar
des lettre apres lettre, mais   mot BASIC (token) apres mot
BASIC. Ils  sont donc en code abrege."
11180 END
11190 REM NEXT missing
11200 IF ERR=26 THEN PRINT "Une boucle 'FOR...";:PEN 3:PRINT
 "NEXT";:PEN 1:PRINT "' se compose de" ELSE GOTO 11250
11210 PRINT "'FOR' et NEXT. Pour la boucle en ligne";ERL;","
11220 PRINT "il manque visiblement un NEXT!           Verifie
z et ajoutez-le ou il convient."
11230 END
11240 REM File already open
11250 IF ERR=27 THEN PRINT "Un fichier qui a deja ete ouvert
 avec   l'instruction ";:PEN 3:PRINT "OPEN":PEN 1 ELSE GOTO
11300
11260 PRINT "ne doit pas etre a nouveau ouvert        en lign
e";ERL;"avec l'instruction":PRINT"OPENIN";
11270 PRINT " ou OPENOUT.":PRINT"Soit vous avez oublie que c
e fichier a  deja ete ouvert dans une autre partie duprogram
```

```
me, soit vous avez relance le    programme avant d'avoir fer
me les"
11280 PRINT"fichiers (avec CLOSEIN, CLOSEOUT) qui   avaient
ete ouverts par le programme    (avec OPENIN, OPENOUT)."
11290 REM Unknown command
11300 IF ERR=28 THEN PRINT "Une instruction en ligne";ERL EL
SE GOTO 11330
11310 PRINT "est n'est pas connue de l'ordinateur    dans ce
tte fonction (";:PEN 3:PRINT "UNKNOWN COMMAND";:PEN 1:PRINT
")."
11320 REM WEND missing
11330 IF ERR=29 THEN PRINT "WHILE...";:PEN 3:PRINT "WEND";:P
EN 1:PRINT " constitue certainement" ELSE GOTO 11380
11340 PRINT "un bon moyen de realiser des programmes structu
res, mais vous avez oublie        jusqu'en ligne";ERL;"de fer
mer dans";:PRINT"votre programme une boucle"
11350 PRINT "WHILE...WEND avec ";:PEN 3:PRINT "WEND";:PEN 1:
PRINT "."
11360 END
11370 REM Unexpected WEND
11380 IF ERR=30 THEN PRINT "WHILE...";:PEN 3:PRINT "WEND";:P
EN 1:PRINT " constitue certainement" ELSE END
11390 PRINT "un bon moyen de realiser des programmes structu
res, mais vous avez oublie        jusqu'en ligne";ERL;"d'ouvr
ir dans";:PRINT"votre programme une boucle"
11400 PRINT "WHILE...WEND avec ";:PEN 3:PRINT "WHILE";:PEN 1
:PRINT "."
11410 END
11420 PEN 3
11430 REM Les messages d'erreur anglais
11440 IF ERR=1 THEN PRINT "Unexpected NEXT";
11450 IF ERR=2 THEN PRINT "Syntax Error";
11460 IF ERR=3 THEN PRINT "Unexpected RETURN";
11470 IF ERR=4 THEN PRINT "DATA exhausted";
11480 IF ERR=5 THEN PRINT "Improper argument";
11490 IF ERR=6 THEN PRINT "Overflow";
11500 IF ERR=7 THEN PRINT "Memory full";
11510 IF ERR=8 THEN PRINT "Line does not exist";
11520 IF ERR=9 THEN PRINT "Subscript out of range";
11530 IF ERR=10 THEN PRINT "Array already dimensioned";
```

```
11540 IF ERR=11 THEN PRINT "Division by zero";
11550 IF ERR=12 THEN PRINT "Invalid direct command"
11560 IF ERR=13 THEN PRINT "Type mismatch";
11570 IF ERR=14 THEN PRINT "String space full";
11580 IF ERR=15 THEN PRINT "String too long";
11590 IF ERR=16 THEN PRINT "String expression too complex";
11600 IF ERR=17 THEN PRINT "Cannot CONTinue";
11610 IF ERR=18 THEN PRINT "Unknown user function";
11620 IF ERR=19 THEN PRINT "RESUME missing";
11630 IF ERR=20 THEN PRINT "Unexpected RESUME";
11640 IF ERR=21 THEN PRINT "Direct command";
11650 IF ERR=22 THEN PRINT "Operand missing";
11660 IF ERR=23 THEN PRINT "Line too long";
11670 IF ERR=24 THEN PRINT "EOF met";
11680 IF ERR=25 THEN PRINT "File type error";
11690 IF ERR=26 THEN PRINT "NEXT missing";
11700 IF ERR=27 THEN PRINT "File already open";
11710 IF ERR=28 THEN PRINT "Unknown command";
11720 IF ERR=29 THEN PRINT "WEND missing";
11730 IF ERR=30 THEN PRINT "Unexpected WEND";
11740 PEN 1
11750 PRINT:PRINT
11760 RETURN
```

Références des variables

Nous avons déjà étudié dans le programme Mémoire 4 comment les différents numéros de ligne et la longueur de chaque ligne sont stockées en mémoire (les adresses 368 et 369 contiennent la longueur de la première ligne, les adresses 370 et 371 contiennent le numéro de la première ligne, etc...). Le programme Mémoire 4 nous a en outre fourni un moyen simple de faire rechercher une expression d'un maximum de 6 lettres dans la mémoire de l'ordinateur. L'utilité du programme Mémoire 4 était donc d'une part de nous permettre de connaître un peu mieux la mémoire de l'ordinateur et d'autre part de pouvoir rechercher une séquence alphanumérique dans toute la mémoire.

Le programme 'références des variables' part du même principe mais va encore beaucoup plus loin: il ne recherche pas seulement des séquences de caractères ou de lettres dans la mémoire, il ordonne de plus les expressions ainsi trouvées.

Comme le CPC permet de distinguer des noms de variables comportant jusqu'à 40 caractères, il faut bien sûr faire rechercher en mémoire le nom de la variable tout entier. Notre tâche (et bien sûr avant tout celle du CPC) est considérablement facilitée par le fait que celui-ci stocke les premiers caractères d'un nom de variable normalement mais que le dernier caractère du nom d'une variable reçoit un code spécial correspondant au code ASCII de ce caractère + 128. Si le nom de la variable ne se compose que d'un seul caractère, c'est bien sûr le premier mais aussi dernier caractère du nom de cette variable qui reçoit ce code spécial

(code ASCII + 128). Le stockage des variables et le listage des références des variables par notre programme tiennent également compte de l'existence de différents types de variables (alphanumériques, nombres entiers, ...).

Voyons maintenant quelles sont les possibilités exactes de notre programme de références des variables:

1) Recherche des variables du programme (le programme 'références des variables commençant en ligne 10 000, la dernière ligne de votre programme doit être au plus 9999).

2) Si le nom de variable trouvé figure au premier rang dans l'ordre alphabétique des variables de votre programme, le programme recherche dans la mémoire programme dans quelles lignes de votre programme cette variable apparaît. C'est ainsi qu'apparaît une liste des références de vos variables qui peut vous être d'un très grand secours dans votre travail de débuggage (recherche des erreurs) de vos programmes BASIC.

Vous pouvez essayer notre programme immédiatement en entrant en lignes 10 à 50 les affectations de variables suivantes:

```
10 a$="CPC464"
20 b$="Ordinateur"
30 CPC=464
40 DATABECKER!=464
50 PROGRAMMESBASIC%=464
```

Vous pouvez lancer maintenant la routine de références des

variables avec RUN 10 000 et vous verrez peu de temps après sur l'écran non seulement la liste des variables utilisées mais en outre les numéros de ligne où ces variables apparaissent.

Une fois que la totalité du programme a été examinée, le CPC indique combien d'octets votre programme occupe en mémoire, sans tenir compte bien sûr du programme de références des variables qu'il vaut mieux que vous supprimiez lorsque votre programme sera complètement au point. Il vous suffira pour celà d'utiliser l'instruction: DELETE 10000-.

Un dernier conseil:

Plus votre programme est long et plus votre CPC mettra de temps à rechercher toutes les variables que vous avez utilisées. Nous vous recommandons donc de faire preuve d'un peu de patience.

Les noms des variables doivent en effet être tout d'abord triés (non seulement d'après l'ordre alphabétique mais aussi en distinguant les majuscules et les minuscules), puis le numéro de la ligne ou des lignes où elles apparaissent doit ou doivent être recherchés et il faut enfin que l'ordinateur affiche toutes ces données à l'écran.

```
10000 REM Liste et references des
10001 '      variables
10010 REM Programmes Basic pour CPC464
10020 REM Copyright 1984 DATA BECKER &
10021 '      Rainer Lueers
10030 CLEAR:DIM a$(1000)
10040 REM Determiner la plus petite et
10041 '      la plus grande variable
10042 '      possibles
10050 marque$=STRING$(40,"z"):marque1$=STRING$(40,"!")
10060 MODE 1
10070 z=368:a$="###":b$="#####"
10080 REM Calcul de la longueur et du
10081 '      numero de la ligne
10090 a=PEEK(z)+256*PEEK(z+1):b=PEEK(z+2)+256*PEEK(z+3)
10100 REM Le calcul ne concerne que la
10101 '      partie du programme figurant
10102 '      avant la ligne 10000
10110 IF b<>10000 THEN GOTO 10150
10120 GOSUB 10360
10130 PRINT:PRINT "Taille du programme:  ";USING b$;z-368;:P
RINT " Octets":PRINT:END
10140 REM Debut de la recherche des
10141 '      varables en memoire
10150 FOR z1=z+4 TO z+a:IF (PEEK(z1)=13 OR (PEEK(z1)>1 AND P
EEK(z1)<5)) AND PEEK(z1+2)=0 THEN GOSUB 10200
10160 IF z+4<>z+a THEN NEXT z1
10170 z=z+a
10180 GOTO 10090
10190 REM Quel type de variables?
10191 '      13 = nombre normal
10192 '      3  = chaine de caracteres
10193 '      2  = nombre entier avec '%'
10194 '      4  = nombre normal avec '!'
10200 IF PEEK(z1)=13 THEN zz=13
10210 IF PEEK(z1)=3 THEN zz=3
10220 IF PEEK(z1)=2 THEN zz=2
10230 IF PEEK(z1)=4 THEN zz=4
10240 REM Examiner longueur du nom
10241 '      de la variable
```

```
10250 IF PEEK(z1+3)>128 THEN zz$=zz$+CHR$((PEEK(z1+3)-128))
ELSE zz$=zz$+CHR$(PEEK(z1+3)):z1=z1+1:GOTO 10250
10260 REM Quel type de variables?
10261 '       13 = nombre normal
10262 '       3  = chaine de caracteres
10263 '       2  = nombre entier avec '%'
10264 '       4  = nombre normal avec '!'
10270 IF zz=13 THEN zz$=zz$
10280 IF zz=3 THEN zz$=zz$+"$"
10290 IF zz=2 THEN zz$=zz$+"%"
10300 IF zz=4 THEN zz$=zz$+"!"
10310 REM Les variables sont rangees
10311 '       dans l'ordre alphabetique
10312 '       avant d'etre listees
10320 IF zz$<marque$ AND zz$>marque1$ THEN marque$=zz$
10330 IF flag=1 AND zz$=a$(compteur) THEN PRINT b;
10340 zz$="":RETURN
10350 REM Ranger le nom de la variable
10351 '       et le numero de la ligne ou
10352 '       il apparait
10360 IF flag=1 THEN flag=0:PRINT:GOTO 10070 ELSE flag=1:IF
marque$=STRING$(40,"z") THEN RETURN ELSE PRINT marque$;:comp
teur=compteur+1:a$(compteur)=marque$:marque1$=marque$:marque
$=STRING$(40,"z")
10370 GOTO 10070
```

Calendrier

==========

Tous ceux qui ont l'habitude de travailler avec des ordinateurs rencontrent fréquemment des programmes de calendrier sous telle ou telle forme. Il s'agit aussi bien de programmes de "calendrier" isolés comme dans le cas présent ou bien de programmes plus important comportant un sous-programme de calendrier (par exemple les programmes d'agenda dont vous trouverez également un listing dans le présent ouvrage).

Il s'agit de toute façon toujours d'additionner les jours, les mois et les années pour calculer comme résultat final quel jour de la semaine sera la date voulue.

Le programme que nous vous proposons comprend trois parties qui résultent de l'entrée avec INPUT de l'année, du mois et du jour calendaire voulus.

L'année entrée doit être comprise entre 1901 et 2000 et les années bissextiles sont calculées automatiquement (ligne 140).

Pour le mois, le nom du mois et le nombre de jours sont déterminés en fonction du numéro de mois entré. Si l'année est une année bissextile (z=1) le 29ème jour du mois de février est pris en compte (ligne 590).

La ligne 750 permet enfin le branchement du programme sur les différents jours de la semaine et l'affichage correspondant (le ... de l'année ... est/était un ...) et si vous le souhaitez vous pouvez faire calculer le jour de la semaine d'une autre date.

```
10 REM Calendrier
20 REM Programmes Basic pour le CPC464
30 REM Copyright 1984 DATA BECKER &
31 '      Rainer Lueers
40 INK 0,1:INK 1,24:INK 2,1,24:effet=2:normal=1
50 MODE 1
60 PRINT
70 PRINT
80 PRINT
90 PRINT "  Determination du jour de la semaine
  de 1901 a 2000"
100 PRINT:PRINT
110 REM Entree de l'annee et calcul
111 '      (pour les annees bissextiles)
120 INPUT "Annee ";j
130 IF j<1901 OR j>2000 THEN GOSUB 990:GOTO 50
140 IF INT(j/4)=j/4 THEN z=1 ELSE z=0
150 j1=2
160 IF j=1901 THEN GOTO 230
170 FOR n=1902 TO j
180 j1=j1+1
190 IF INT((n-1)/4)=(n-1)/4 THEN j1=j1+1
200 IF j1>6 THEN j1=j1-7
210 NEXT n
220 REM Entree du mois et calcul
221 '      (nom du mois et nombre de jours)
230 INPUT "Mois ";m
240 IF m<1 OR m>12 THEN GOSUB 990:GOTO 50
250 ON m GOSUB 270,290,310,330,350,370,390,410,430,450,470,4
90
260 GOTO 510
270 m$="Janvier"
280 RETURN
290 m$="Fevrier"
300 RETURN
310 m$="Mars"
320 RETURN
330 m$="Avril"
340 RETURN
350 m$="Mai"
```

```
360 RETURN
370 m$="Juin"
380 RETURN
390 m$="Juillet"
400 RETURN
410 m$="Aout"
420 RETURN
430 m$="Septembre"
440 RETURN
450 m$="Octobre"
460 RETURN
470 m$="Novembre"
480 RETURN
490 m$="Decembre"
500 RETURN
510 RESTORE:FOR n=1 TO m
520 READ m1
530 NEXT n
540 DATA 31,28,31,30,31,30
550 DATA 31,31,30,31,30,31
560 IF z=1 AND m=2 THEN m1=29
570 REM Entree du jour (numero) et calcul
571 '      du jour de la semaine
580 INPUT "Jour ";t
590 IF t=29 AND m=2 AND z<>1 THEN GOSUB 990:GOTO 50
600 IF t<1 OR t>m1 THEN GOSUB 990:GOTO 50
610 t1=0
620 RESTORE
630 IF m=1 THEN GOTO 690
640 FOR n=2 TO m
650 READ m1
660 t1=t1+m1
670 NEXT n
680 IF z=1 AND m>2 THEN t1=t1+1
690 t1=t1+t+j1
700 t1=t1-INT(t1/7)*7
710 IF t1=0 THEN t1=7
720 PRINT
730 PRINT "      Le";t;m$;" de l'annee";j
740 PRINT "           est/etait un ";
```

```
750 ON t1 GOSUB 840,860,880,900,920,940,960
760 PRINT
770 PRINT
780 INPUT " Voulez-vous calculer          un autre jour de l
a semaine ( /N) ";f$
790 f$=UPPER$(f$)
800 IF f$="N" THEN END
810 RESTORE
820 CLS
830 GOTO 60
840 PRINT"Dimanche"
850 RETURN
860 PRINT"Lundi"
870 RETURN
880 PRINT"Mardi"
890 RETURN
900 PRINT"Mercredi"
910 RETURN
920 PRINT"Jeudi"
930 RETURN
940 PRINT"Vendredi"
950 RETURN
960 PRINT"Samedi"
970 RETURN
980 END
990 PEN effet:PRINT:PRINT TAB(12) "Entree incorrecte!"
1000 PEN normal:GOSUB 1010:RETURN
1010 PRINT:PRINT TAB(7) "<Appuyez sur une touche S.V.P.>"
1020 f$=INKEY$:IF f$="" THEN GOTO 1020
1030 RETURN
```

Fichier de disques

=================

Ce programme ne méritera son nom que si vous le tapez mot pour mot, instruction pour instruction, exactement comme il figure dans notre recueil de programmes.

Mais comme le produit final, moyennant cette précaution, sera une petite banque de données aux performances pas du tout négligeables, il est aussi possible de modifier certaines instructions PRINT, de façon par exemple à ce que PRINT"Entrez les disques" devienne PRINT"Entrez les adresses.

Vous pourriez ainsi vous constituer très facilement, en plus de votre programme de fichier de disques, un programme de fichier d'adresses qu'il faudrait bien sûr que vous sauvegardiez séparément.

Nous reconnaissons toutefois que le programme de fichier de disques pâtit malgré tout de certaines contraintes qui ne pourraient être valablement éliminées que si nous ne travaillions pas sur cassette mais sur disquette.

La première limite du programme tient bien sûr à la capacité mémoire de notre ordinateur: si nous utilisons pleinement une fiche (trois fois 40 caractères), nous occupons déjà 120 octets en mémoire. 100 fiches représentent donc 12000 octets. C'est pourquoi nous avons dû limiter à 300 le nombre maximum de fiches différentes.

Si vous êtes certain que vous n'utiliserez pas autant de caractères par fiche, vous pouvez modifier le programme de façon à remonter le nombre limite de fiches. Vous pouvez par exemple modifier ainsi la ligne 100:

```
100 IF z<1 OR z>400...
```

Le programme 'fichier de disques' vous offre les possibilités suivantes:

1. Vous pouvez constituer des fiches de trois champs (nom du disque, chanteur, numéro).

2. Vous pouvez charger des fichiers à partir d'une cassette.

3. Vous pouvez sauvegarder des fichiers sur cassette.

4. Vous pouvez faire rechercher un disque en mémoire (d'après le nom du disque, du chanteur ou le numéro). Pour cette fonction, l'écran affiche les données trouvées qui correspondent à ce que vous cherchez jusqu'à ce que l'écran soit plein. Vous appuyez alors sur ENTER pour que la recherche se poursuive. Ceci vous permet donc de faire rechercher par exemple dans votre fichier tous les disques d'un chanteur que vous possédez.

5. Vous pouvez faire éditer la liste complète de vos disques sur l'écran ou sur une imprimante (pour l'affichage à l'écran, les données sont affichées par groupe de 5 et vous devez appuyer sur la touche ENTER pour passer à la suite).

6. Vous pouvez modifier les données entrées. Comme la routine d'entrée (point 1) vous permet déjà de corriger les données que vous avez entrées, il est peu probable que vous utiliserez cette fonction pour corriger le nom des chanteurs ou des disques figurant dans votre fichier. Nous avons en fait surtout pensé ici au cas où vous transformeriez ce programme en programme de fichier d'adresses. Il arrive en effet fréquemment qu'on ait à modifier les données d'un fichier d'adresses à la suite de changements d'adresses. Il vous suffira alors de sélectionner le point 6 du programme. Vous pourrez ensuite laisser inchangées certaines informations en appuyant simplement sur ENTER et en modifier d'autres en entrant les nouvelles informations correspondantes.

Que vous utilisiez ce programme pour vos disques ou que vous le transformiez en programme de fichier d'adresses, nous vous souhaitons en tout cas de bons moments et nous espérons qu'il vous permettra de mettre un peu d'ordre dans vos affaires.

```
10 REM Fichier de disques
20 REM Programmes Basic pour le CPC464
30 REM Copyright 1984 DATA BECKER &
31 '      Rainer Lueers
40 INK 0,1:INK 1,24:INK 2,1,24:effet=2:normal=1
50 m=1:MODE 1
60 REM Preparatifs avant dimensionnement
61 '      des variables
70 PAPER 1:PEN 0:PRINT "Fichier de disques":PAPER 0:PEN 1
80 PRINT
90 INPUT "Combien de disques voulez-vous          enregistre
r (maximum 300) ";z
100 IF z<1 OR z>300 THEN GOSUB 2300:GOTO 50
110 fr=FRE(0):DIM p$(z),s$(z),n$(z)
120 PRINT:PRINT "Le seul dimensionnement a";z:PRINT "disques
 necessite";fr-FRE(0);"cases memoire."
130 PRINT:PRINT:PRINT "Restent donc par disque au maximum:":
PRINT:PRINT "            ";INT(FRE(0)/z):PRINT:PRINT "cases m
emoire disponibles.":PRINT:PRINT:PRINT
140 INPUT "Est-ce suffisant (sinon il faut changer le dimens
ionnement des disques a entrer)=  ( /N) ";f$
150 f$=LEFT$(UPPER$(f$),1)
160 IF f$="N" THEN CLEAR:GOTO 50
170 REM Affichage du menu principal
180 CLS
190 LOCATE 6,8
200 PAPER 1:PEN 0:PRINT "Fichier de disques":PAPER 0:PEN 1
210 LOCATE 16,10
220 PRINT "Menu"
230 LOCATE 8,13
240 PRINT "1.Entrer disques"
250 LOCATE 8,14:PRINT "2.Charger disques"
260 LOCATE 8,15:PRINT "3.Sauvegarder disques"
270 LOCATE 8,16:PRINT "4.Rechercher disques"
280 LOCATE 8,17:PRINT "5.Sortir disques"
290 LOCATE 8,18:PRINT "6.Modifier disques"
300 LOCATE 6,21
310 INPUT "Votre choix (1/2/3/4/5/6) ";f$
320 IF f$<"1" OR f$>"6" THEN GOSUB 2300:GOTO 180
330 ON VAL(f$) GOSUB 370,680,940,1220,1750,2020
```

```
340 f$="":f1$="":f2$="":n1=0
350 GOTO 180
360 REM Sous-programme d'entree des
361 '      donnees
370 FOR n=m TO z
380 CLS
390 PAPER 1:PEN 0:PRINT "1.Entrer des disques":PAPER 0:PEN 1
400 PRINT
410 GOSUB 2360
420 PRINT
430 f1$="":PRINT "Nom du disque"
440 INPUT p$(n):IF LEN(p$(n))>40 THEN p$(n)=LEFT$(p$(n),40)
450 IF p$(n)="" THEN m=n:RETURN
460 PRINT "Chanteur, Groupe:"
470 INPUT s$(n):IF LEN(s$(n))>40 THEN s$(n)=LEFT$(s$(n),40)
480 PRINT "Numero/Code:"
490 INPUT n$(n):IF LEN(n$(n))>40 THEN n$(n)=LEFT$(n$(n),40)
500 REM Premier affichage des donnees
501 '      entrees avec premiere
502 '      possibilite de modification
510 PRINT
520 PRINT "Voici encore une fois les donnees       entrees:"
530 PRINT "Numero d'enregistrement";n;"sur 1 a";z
540 PRINT
550 PAPER 1:PEN 0:PRINT "Nom du disque:":PAPER 0:PEN 1
560 PRINT p$(n)
570 PAPER 1:PEN 0:PRINT "Chanteur, Groupe:":PAPER 0:PEN 1
580 PRINT s$(n)
590 PAPER 1:PEN 0:PRINT "Numero/Code:":PAPER 0:PEN 1
600 PRINT n$(n)
610 PRINT
620 INPUT "Correct ( /N)";f1$
630 f1$=UPPER$(f1$)
640 IF f1$="N" THEN PRINT:GOTO 430
650 NEXT n
660 RETURN
670 REM Sous-programme de chargement de
671 '      fichiers sur disquette
680 CLS
690 PAPER 1:PEN 0:PRINT "2.Charger des disques":PAPER 0:PEN
```

```
1
700 PRINT
710 GOSUB 2360
720 PRINT
730 PRINT"En chargeant un fichier, vous detruisez les donnee
s pouvant etre en memoire!"
740 PRINT
750 INPUT "Voulez-vous charger des donnees (O/ )";f1$
760 f1$=LEFT$(UPPER$(f1$),1)
770 IF f1$="O" THEN GOTO 780 ELSE RETURN
780 PRINT
790 PRINT "Nom du fichier"
800 INPUT dn$
810 dn$="!"+dn$
820 OPENIN dn$
830 INPUT #9,m
840 PRINT
850 PRINT "Le fichier a charger ";dn$
860 PRINT "se compose de";m-1;"enregistrements"
870 FOR n=1 TO m-1
880 INPUT #9,p$(n)
882 INPUT #9,s$(n)
884 INPUT #9,n$(n)
890 PRINT n:PRINT p$(n):PRINT s$(n):PRINT n$(n)
900 NEXT n
910 CLOSEIN
920 RETURN
930 REM Sous-programme de sauvegarde de
931 '      fichiers sur disquette
940 CLS
950 PAPER 1:PEN 0:PRINT "3.Sauvegarder des disques":PAPER 0:
PEN 1
960 PRINT
970 GOSUB 2360
980 PRINT
990 INPUT "Voulez-vous sauvegarder des donnees     (O/ ) ";f
1$
1000 f1$=UPPER$(f1$)
1010 IF f1$="O" THEN GOTO 1020 ELSE RETURN
1020 PRINT
```

```
1030 IF dn$<>"" THEN PRINT "Faut-il le sauvegarder sous le n
om":PRINT dn$;:INPUT " (O/ ) ";f2$:f2$=LEFT$(UPPER$(f2$),1):
IF f2$="O" THEN GOTO 1070
1040 PRINT "Nom du fichier ";
1050 INPUT dn$
1060 dn$="!"+dn$
1070 OPENOUT dn$
1080 PRINT #9,m
1090 PRINT
1100 PRINT "Le fichier a sauvegarder ";dn$
1105 PRINT "se compose de";m-1;"enregistrements"
1110 FOR n=1 TO m-1
1120 PRINT #9,p$(n)
1122 PRINT #9,s$(n)
1124 PRINT #9,n$(n)
1130 PRINT n
1140 PRINT p$(n)
1150 PRINT s$(n)
1160 PRINT n$(n)
1170 PRINT
1180 NEXT n
1190 CLOSEOUT
1200 RETURN
1210 REM Sous-programme de recherche de
1211 '      disques, chanteurs ou numeros
1220 CLS
1230 PAPER 1:PEN 0:PRINT "4.Recherche de disques":PAPER 0:PE
N 1
1240 PRINT
1250 GOSUB 2360
1260 PRINT
1270 REM Affichage d'un sous-menu:
1271 '      Recherche d'apres:
1272 '      1) Nom du disque
1273 '      2) Chanteur, Groupe
1274 '      3) Numero/Code
1280 PRINT "Recherche d'apres quel critere:"
1290 PRINT
1300 PRINT "1.Nom du disque"
1310 PRINT"2.Chanteur, Groupe"
```

```
1320 PRINT "3.Numero/Code"
1330 PRINT
1340 INPUT "Votre choix (1/2/3) ";f1$
1350 IF f1$<"1" OR f1$>"3" THEN RETURN
1360 ON VAL(f1$) GOSUB 1390,1510,1620
1370 f1$="":f2$="":n1=0:GOTO 1220
1380 REM Recherche d'apres le nom du
1381 '      disque
1390 CLS
1400 PRINT "Nom du disque"
1410 INPUT p$:IF LEN(p$)>40 THEN p$=LEFT$(p$,40)
1420 IF p$="" THEN RETURN
1430 FOR n=1 TO m-1
1440 IF p$=p$(n) THEN PRINT n:PRINT p$(n):PRINT s$(n):PRINT
n$(n):n1=n1+1
1450 IF n1=5 THEN n1=0:GOSUB 2330
1460 NEXT n
1470 PRINT
1480 GOSUB 2330
1490 RETURN
1500 REM Recherche d'apres chanteur,
1501 '      groupe
1510 CLS
1520 PRINT "Chanteur, Groupe"
1530 INPUT s$:IF LEN(s$)>40 THEN s$=LEFT$(s$,40)
1540 IF s$="" THEN RETURN
1550 FOR n=1 TO m-1
1560 IF s$=s$(n) THEN PRINT n:PRINT p$(n):PRINT s$(n):PRINT
n$(n):n1=n1+1
1570 IF n1=5 THEN n1=0:GOSUB 2330
1580 NEXT n
1590 PRINT
1600 GOSUB 2330
1610 RETURN
1620 CLS
1630 REM Recherche d'apres numero/code
1640 PRINT "Numero/Code"
1650 INPUT n$
1660 IF n$="" THEN RETURN
1670 FOR n=1 TO m-1
```

```
1680 IF n$=n$(n) THEN PRINT n:PRINT p$(n):PRINT s$(n):PRINT
n$(n):n1=n1+1
1690 IF n1=5 THEN n1=0:GOSUB 2330
1700 NEXT n
1710 PRINT
1720 GOSUB 2330
1730 RETURN
1740 REM Sous-programme de sortie des
1741 '      donnees sauvegardees sur
1742 '      ecran ou imprimante
1750 CLS
1760 PAPER 1:PEN 0:PRINT "5.Sortir disques":PAPER 0:PEN 1
1770 PRINT
1780 GOSUB 2360
1790 PRINT
1800 PRINT "Sortie sur ecran ou sur imprimante?"
1810 INPUT "(E/I)";f2$
1820 IF f2$="" THEN RETURN
1830 f2$=LEFT$(UPPER$(f2$),1)
1840 IF f2$="I" THEN GOTO 1950
1850 REM Sortie des donnees sur le
1851 '      moniteur
1860 CLS:FOR n=1 TO m-1
1870 n1=n1+1
1880 IF n1=5 THEN n1=0:GOSUB 2330:CLS
1890 PRINT n:PRINT p$(n):PRINT s$(n):PRINT n$(n)
1900 NEXT n
1910 PRINT
1920 GOSUB 2330
1930 RETURN
1940 REM Sortie des donnees sur
1941 '      imprimante
1950 FOR n=1 TO m-1
1960 PRINT #8,n:PRINT #8,p$(n):PRINT #8,s$(n):PRINT #8,n$(n)
:PRINT #8
1970 NEXT n
1980 PRINT
1990 GOSUB 2330
2000 RETURN
2010 REM Sous-programme de modification
```

```
2011 '     des donnees en memoire
2020 CLS
2030 PAPER 1:PEN 0:PRINT "6.Modification des disques":PAPER
0:PEN 1
2040 PRINT
2050 GOSUB 2360
2060 PRINT
2070 INPUT "Numero du disque ";n1
2080 IF n1>m-1 OR n1<0 THEN GOSUB 2300:GOTO 2070
2090 IF n1=0 THEN RETURN
2100 PRINT
2110 PRINT n1:PRINT p$(n1):PRINT s$(n1):PRINT n$(n1)
2120 PRINT
2130 PRINT "Faut-il changer (Non=<ENTER>)?"
2140 PRINT p$(n1)
2150 INPUT d$:IF d$<>"" THEN p$(n1)=d$
2160 PRINT s$(n1)
2170 INPUT d$:IF d$<>"" THEN s$(n1)=d$
2180 PRINT n$(n1)
2190 INPUT d$:IF d$<>"" THEN n$(n1)=d$
2200 CLS
2210 PRINT "Entree pour numero de disque";n1;":"
2220 PRINT p$(n1)
2230 PRINT s$(n1)
2240 PRINT n$(n1)
2250 PRINT
2260 GOSUB 2330
2270 n1=0
2280 GOTO 2020
2290 REM Sous-programme pour message
2291 '     quand entree incorrecte
2300 PRINT:PEN effet:PRINT TAB(12) "Entree incorrecte!":PEN
normal:GOSUB 2330:RETURN
2310 a$=INKEY$:IF a$="" THEN GOTO 2310 ELSE RETURN
2320 REM Sous-programme pour geler
2321 '     l'affichage a l'ecran
2330 PRINT:PRINT TAB(7) "<Appuyez sur une touche S.V.P.>"
2340 a$=INKEY$:IF a$="" THEN GOTO 2340 ELSE RETURN
2350 REM Sous-programme general
2360 PRINT "Retour au menu avec <ENTER>, sans entreraucun mo
```

```
t"
2370 RETURN
```

Tableau sportif
===============

Ce programme peut vous réserver de très bons moments et vous faire en outre gagner beaucoup de temps si vous vous intéressez au sport.

Votre ordinateur va en effet vous aider à gérer les nombreux évènements réguliers qui émaillent une année sportive. Nous avons choisi comme exemple le football qui a donc servi de référence pour notre programme.

Au début de la routine d'entrée, vous pouvez choisir de gérer jusqu'à 20 clubs. Le CPC vous demande ensuite si les noms des clubs qui figurent déjà en lignes DATA (clubs de 1ère division) correspondent bien aux clubs en compétition cette saison ou si vous voulez entrer vous-même une liste de jusqu'à 20 clubs. Le fonctionnement de ce programme est très simple: il vous permet d'entrer les résultats de chaque journée dès qu'ils sont connus et de voir alors le tableau qui se dégage de ces résultats. Vous pouvez consulter les résultats de toutes les journées que vous avez déjà entrées et bien sûr vous pouvez également sauvegarder les résultats sur cassette.

Vous pouvez entrer la liste des clubs et le calendrier en mode direct ou les modifier dans les lignes de DATA. Vous pourrez ensuite recharger à tous moments toutes les données que vous avez déjà sauvegardées sur cassette, en choisissant l'option charger un tableau en début de programme. Nous vous souhaitons un grand championnat!

```
10 REM Tableau sportif
20 REM Programmes Basic pour le CPC 464
30 REM Copyright 1984 DATA BECKER &
31 '      Rainer Lueers
40 INK 0,1:INK 1,26:INK 2,1,24:effet=2:normal=1:PEN 1:PAPER
0
50 MODE 1
54 INPUT "Charger un tableau (O/ )";f$
56 MODE 1:f$=UPPER$(f$):IF LEFT$(f$,1)="O" THEN GOSUB 7000:G
OTO 530
60 REM Choix du nombre de clubs
61 '      (de 4 a 20)
70 INPUT "Combien de clubs (4/6/8/10/12/14/16/18  ou 20) ";f
$
80 REM Liste des variables qui sont
81 '      dimensionnee avec leurs
82 '      significations respectives
90 REM a$( )=nom de club
91 '    b( , )=rencontre ayant eu lieu
92 '    c( , , )=resultat
93 '    e( , )=total buts
94 '    f( , )=total buts encaisses
100 REM g( , )=total matchs gagnes
101 '    h( , )=total matchs nuls
102 '    i( , )=total matchs perdus
103 '    k( , , )=rencontres jumelees
104 '    z1 bis z5=pour routine de tri
110 f=VAL(f$):IF f/2<>INT(f/2) THEN PRINT:GOSUB 1290:GOTO 50
120 IF f<4 OR f>20 THEN GOSUB 1290:GOTO 50
125 GOSUB 130:GOTO 140
130 DIM a$(f),b(f*2,f/2),c(f,f*2-2,2),e(f),f(f),g(f),h(f),i(
f),k(f-1,f/2,2),z1(f+1),z2(f+1),z3(f+1),z4(f+1),z5(f+1):RETU
RN
140 REM Entree directe de noms de clubs
141 '      ou entree des noms de clubs
142 '      a partir des lignes de DATA
150 INPUT "Noms des lignes de programme (O/ )";f$
160 f$=UPPER$(f$):IF LEFT$(f$,1)="O" THEN GOTO 230
170 FOR n=1 TO f
180 PRINT "Club";n;:INPUT ": ";f$
```

```
190 IF f$="" THEN GOTO 180
200 IF LEN(f$)>10 THEN a$(n)=LEFT$(f$,10) ELSE a$(n)=f$
210 NEXT n
220 GOTO 350
230 ON f GOSUB 250,250,250,250,250,260,250,270,250,280,250,2
90,250,300,250,310,250,320,250,330
240 GOTO 350
250 RESTORE 6350:FOR n=1 TO f:READ a$(n):NEXT n:RETURN
260 RESTORE 6320:FOR n=1 TO f:READ a$(n):NEXT n:RETURN
270 RESTORE 6290:FOR n=1 TO f:READ a$(n):NEXT n:RETURN
280 RESTORE 6260:FOR n=1 TO f:READ a$(n):NEXT n:RETURN
290 RESTORE 6230:FOR n=1 TO f:READ a$(n):NEXT n:RETURN
300 RESTORE 6200:FOR n=1 TO f:READ a$(n):NEXT n:RETURN
310 RESTORE 6170:FOR n=1 TO f:READ a$(n):NEXT n:RETURN
320 RESTORE 6130:FOR n=1 TO f:READ a$(n):NEXT n:RETURN
330 RESTORE 6080:FOR n=1 TO f:READ a$(n):NEXT n:RETURN
340 REM Le calendrier en lignes de DATA
341 '      (systeme habituel) peut etre
342 '      utilise, mais il est egalement
343 '      possible d'entrer son propre
344 '      calendrier
350 INPUT "Calendrier en lignes DATA (O/ )";f$:f$=UPPER$(f$)
:IF LEFT$(f$,1)="O" THEN GOTO 420
360 PRINT:PRINT:PRINT "Veuillez entrer le calendrier avec de
s  chiffres (Exemple: 1<virgule>2<ENTER>)  O pour renouveler
 derniere entree"
370 FOR n=1 TO f-1:PRINT:PRINT:PRINT "Journee";n:PRINT
380 FOR m=1 TO f/2
390 PRINT:PRINT "Match";m:INPUT a,b:IF a>f OR b>f OR a=b THE
N GOTO 390 ELSE IF a=0 OR b=0 THEN m=m-1:GOTO 390 ELSE k(n,m
,1)=a:k(n,m,2)=b:a=0:b=0:PRINT a$(k(n,m,1));" - ";a$(k(n,m,2
))
400 NEXT m,n
410 GOTO 530
420 ON f GOSUB 440,440,440,440,440,450,440,460,440,470,440,4
80,440,490,440,500,440,510,440,520
430 GOTO 530
440 RESTORE 5000:FOR n=1 TO f-1:FOR m=1 TO f/2:READ k(n,m,1)
,k(n,m,2):NEXT m,n:RETURN
450 RESTORE 5040:FOR n=1 TO f-1:FOR m=1 TO f/2:READ k(n,m,1)
```

```
,k(n,m,2):NEXT m,n:RETURN
460 RESTORE 5100:FOR n=1 TO f-1:FOR m=1 TO f/2:READ k(n,m,1)
,k(n,m,2):NEXT m,n:RETURN
470 RESTORE 5180:FOR n=1 TO f-1:FOR m=1 TO f/2:READ k(n,m,1)
,k(n,m,2):NEXT m,n:RETURN
480 RESTORE 5280:FOR n=1 TO f-1:FOR m=1 TO f/2:READ k(n,m,1)
,k(n,m,2):NEXT m,n:RETURN
490 RESTORE 5400:FOR n=1 TO f-1:FOR m=1 TO f/2:READ k(n,m,1)
,k(n,m,2):NEXT m,n:RETURN
500 RESTORE 5540:FOR n=1 TO f-1:FOR m=1 TO f/2:READ k(n,m,1)
,k(n,m,2):NEXT m,n:RETURN
510 RESTORE 5700:FOR n=1 TO f-1:FOR m=1 TO f/2:READ k(n,m,1)
,k(n,m,2):NEXT m,n:RETURN
520 RESTORE 5880:FOR n=1 TO f-1:FOR m=1 TO f/2:READ k(n,m,1)
,k(n,m,2):NEXT m,n:RETURN
530 MODE 1:PEN effet:LOCATE 7,7:PRINT "Tableau des clubs";:P
EN normal
540 LOCATE 7,10
550 PRINT "1.Entrer journee"
560 LOCATE 7,11
570 PRINT "2.Afficher tableau actuel"
580 LOCATE 7,12
590 PRINT "3.Examiner journee"
592 LOCATE 7,13
594 PRINT "4.Sauvegarder tableau actuel"
600 LOCATE 7,15
610 INPUT "Votre choix (1 a 4) ";f$
620 IF VAL(f$)<1 OR VAL(f$)>4 THEN GOSUB 1270:GOSUB 1290:GOT
O 420
630 ON VAL(f$) GOSUB 660,900,1010,8000
640 GOTO 530
650 REM Entrer journee
660 MODE 1
670 INPUT "Quelle journee ";f$:IF VAL(f$)<1 OR VAL(f$)>f*2-2
 THEN GOSUB 1290:GOTO 670
680 IF VAL(f$)>f-1 THEN GOTO 740
690 FOR n=1 TO f/2:PRINT "Match";n
700 IF b(VAL(f$),n)<>0 THEN PRINT CHR$(7);"Erreur! Resultat
connu:":PRINT a$(k(VAL(f$),n,1));"-";a$(k(VAL(f$),n,2));":";
c(k(VAL(f$),n,1),VAL(f$),1);"-";c(k(VAL(f$),n,1),VAL(f$),2):
```

```
GOTO 730
710 PRINT a$(k(VAL(f$),n,1));" - ";a$(k(VAL(f$),n,2));:INPUT
 a,b:IF a=99 THEN PRINT "annule":b(VAL(f$),n)=0:GOTO 730 ELS
E GOSUB 790
720 b(VAL(f$),n)=1:c(k(VAL(f$),n,1),VAL(f$),1)=a:c(k(VAL(f$)
,n,1),VAL(f$),2)=b:c(k(VAL(f$),n,2),VAL(f$),1)=b:c(k(VAL(f$)
,n,2),VAL(f$),2)=a
730 NEXT n:RETURN
740 FOR n=1 TO f/2:PRINT "Match";n
750 IF b(VAL(f$),n)<>0 THEN PRINT CHR$(7);"Erreur! Resultat
connu:":PRINT a$(k(VAL(f$)-f+1,n,2));"-";a$(k(VAL(f$)-f+1,n,
1));":";c(k(VAL(f$)-f+1,n,1),VAL(f$),1);"-";c(k(VAL(f$)-f+1,
n,1),VAL(f$),2):GOTO 780
760 PRINT a$(k(VAL(f$)-f+1,n,2));" - ";a$(k(VAL(f$)-f+1,n,1)
);:INPUT a,b:IF a=99 THEN PRINT "annule":b(VAL(f$),n)=0:GOTO
 780 ELSE GOSUB 840
770 b(VAL(f$),n)=1:c(k(VAL(f$)-f+1,n,1),VAL(f$),1)=b:c(k(VAL
(f$)-f+1,n,1),VAL(f$),2)=a:c(k(VAL(f$)-f+1,n,2),VAL(f$),1)=a
:c(k(VAL(f$)-f+1,n,2),VAL(f$),2)=b
780 NEXT n:RETURN
790 e(k(VAL(f$),n,1))=e(k(VAL(f$),n,1))+a:f(k(VAL(f$),n,1))=
f(k(VAL(f$),n,1))+b:e(k(VAL(f$),n,2))=e(k(VAL(f$),n,2))+b:f(
k(VAL(f$),n,2))=f(k(VAL(f$),n,2))+a
800 IF a=b THEN h(k(VAL(f$),n,1))=h(k(VAL(f$),n,1))+1:h(k(VA
L(f$),n,2))=h(k(VAL(f$),n,2))+1
810 IF a>b THEN g(k(VAL(f$),n,1))=g(k(VAL(f$),n,1))+2:i(k(VA
L(f$),n,2))=i(k(VAL(f$),n,2))+2
820 IF a<b THEN i(k(VAL(f$),n,1))=i(k(VAL(f$),n,1))+2:g(k(VA
L(f$),n,2))=g(k(VAL(f$),n,2))+2
830 RETURN
840 e(k(VAL(f$)-f+1,n,1))=e(k(VAL(f$)-f+1,n,1))+b:f(k(VAL(f$
)-f+1,n,1))=f(k(VAL(f$)-f+1,n,1))+a:e(k(VAL(f$)-f+1,n,2))=e(
k(VAL(f$)-f+1,n,2))+a:f(k(VAL(f$)-f+1,n,2))=f(k(VAL(f$)-f+1,
n,2))+b
850 IF a=b THEN h(k(VAL(f$)-f+1,n,1))=h(k(VAL(f$)-f+1,n,1))+
1:h(k(VAL(f$)-f+1,n,2))=h(k(VAL(f$)-f+1,n,2))+1
860 IF a>b THEN g(k(VAL(f$)-f+1,n,2))=g(k(VAL(f$)-f+1,n,2))+
2:i(k(VAL(f$)-f+1,n,1))=i(k(VAL(f$)-f+1,n,1))+2
870 IF a<b THEN i(k(VAL(f$)-f+1,n,2))=i(k(VAL(f$)-f+1,n,2))+
2:g(k(VAL(f$)-f+1,n,1))=g(k(VAL(f$)-f+1,n,1))+2
```

```
880 RETURN
890 REM Afficher le tableau actuel
900 MODE 2:GOSUB 1180
910 PRINT"Pl. Club         Matchs  +    =    -  Goal average
 <>    Points    Pl."
920 PRINT STRING$(71,"="):FOR n=1 TO f
930 PRINT n;TAB(5);a$(z5(n));TAB(18);(g(z5(n))+(h(z5(n))*2)+
i(z5(n)))/2;TAB(24);g(z5(n))/2;TAB(29);h(z5(n));TAB(34);i(z5
(n))/2;TAB(40);e(z5(n));TAB(45);":";f(z5(n));
940 PRINT TAB(51);e(z5(n))-f(z5(n));TAB(59);g(z5(n))+h(z5(n)
);TAB(68);n
950 NEXT n
960 PRINT STRING$(71,"=")
970 PRINT TAB(21);"<Appuyez sur une touche S.V.P.>"
980 a$=INKEY$:IF a$="" THEN GOTO 980
990 RETURN
1000 REM Examiner journee
1010 CLS:PRINT:INPUT "Toute la saison (O/ ) ";f$:f$=UPPER$(f
$):IF LEFT$(f$,1)="O" THEN GOTO 1100
1020 PRINT "Journee ( 1 a";f*2-2;:INPUT ") ";f$:IF VAL(f$)<1
 OR VAL(f$)>f*2-2 THEN GOSUB 1290:GOTO 1020 ELSE IF VAL(f$)>
f-1 THEN GOTO 1060
1030 n=VAL(f$):CLS:PRINT "Journee";n:PRINT:PRINT:PRINT:FOR m
=1 TO f/2:PRINT m;TAB(5);a$(k(n,m,1));TAB(15);a$(k(n,m,2));
1040 IF b(n,m)=0 THEN PRINT TAB(33);"-  :  -": ELSE PRINT TAB
(32);c(k(n,m,1),n,1);TAB(36);":";TAB(37);c(k(n,m,1),n,2);
1050 NEXT m:PRINT:PRINT:GOSUB 1270:GOTO 1170
1060 n=VAL(f$):CLS:PRINT "Journee";n:n=n-(f-1):PRINT:PRINT:P
RINT:FOR m=1 TO f/2:PRINT m;TAB(5);a$(k(n,m,2));TAB(14);a$(k
(n,m,1));TAB(23);
1070 IF b(n,m)=1 THEN PRINT c(k(n,m,1),n,2);TAB(27);":";TAB(
28);c(k(n,m,1),n,1); ELSE PRINT " -  :  -";
1080 IF b(n+(f-1),m)=0 THEN PRINT TAB(33);"-  :  -" ELSE PRIN
T TAB(32);c(k(n,m,1),n+f-1,1);TAB(36);":";TAB(37);c(k(n,m,1)
,n+f-1,2);
1090 NEXT m:PRINT:PRINT:GOSUB 1270:GOTO 1170
1100 FOR n=1 TO f-1:CLS:PRINT "Journee";n:PRINT:PRINT:PRINT:
FOR m=1 TO f/2:PRINT m;TAB(5);a$(k(n,m,1));TAB(15);a$(k(n,m,
2));
1110 IF b(n,m)=0 THEN PRINT TAB(33);"-  :  -"; ELSE PRINT TAB
```

```
(32);c(k(n,m,1),n,1);TAB(36);":";TAB(37);c(k(n,m,1),n,2);
1120 NEXT m:PRINT:PRINT:GOSUB 1270:NEXT n
1130 FOR n=1 TO f-1:CLS:PRINT "Journee";n+f-1:PRINT:PRINT:PR
INT:FOR m=1 TO f/2:PRINT m;TAB(5);a$(k(n,m,2));TAB(14);a$(k(
n,m,1));TAB(23);
1140 IF b(n,m)=0 THEN PRINT " -  : -"; ELSE PRINT c(k(n,m,1)
,n,2);TAB(27);":";TAB(28);c(k(n,m,1),n,1);
1150 IF b(n+f-1,m)=0 THEN PRINT TAB(33);"-  : -"; ELSE PRINT
 TAB(32);c(k(n,m,1),n+f-1,1);TAB(36);":";TAB(37);c(k(n,m,1),
n+f-1,2);
1160 NEXT m:PRINT:PRINT:GOSUB 1270:NEXT
1170 RETURN
1180 REM Sous-routine de tri
1190  FOR n=1 TO f:z1(n)=g(n)+h(n):z2(n)=e(n)-f(n):z3(n)=e(n
):z4(n)=-1:NEXT n
1200  FOR n=1 TO f
1210  FOR m=1 TO f
1220  IF z1(m)>z4(n) THEN z4(n)=z1(m):a=m:GOTO 1250
1230  IF z1(m)=z4(n) AND z2(m)>z2(a) THEN z4(n)=z1(m):a=m:GO
TO 1250
1240  IF z1(m)=z4(n) AND z2(m)=z2(a) AND z3(m)>z3(a) THEN z4
(n)=z1(m):a=m
1250  NEXT m:z1(a)=-1:z5(n)=a:a=f+1:NEXT n
1260  RETURN
1270 PRINT:PRINT TAB(7);"<Appuyez sur une touche S.V.P.>"
1280 a$=INKEY$:IF a$="" THEN GOTO 1280 ELSE RETURN
1290 PEN effet:PRINT TAB(12) "Entree incorrecte!":PEN normal
:GOSUB 1270:RETURN
5000 REM Datas pour 4 clubs
5010 DATA 1,4,2,3
5020 DATA 4,3,1,2
5030 DATA 2,4,3,1
5040 REM Datas pour 6 clubs
5050 DATA 1,6,2,5,3,4
5060 DATA 6,4,5,3,1,2
5070 DATA 2,6,3,1,4,5
5080 DATA 6,5,1,4,2,3
5090 DATA 3,6,4,2,5,1
5100 REM Datas pour 8 clubs
5110 DATA 1,8,2,7,3,6,4,5
```

```
5120 DATA 8,5,6,4,7,3,1,2
5130 DATA 2,8,3,1,4,7,5,6
5140 DATA 8,6,7,5,1,4,2,3
5150 DATA 3,8,4,2,5,1,6,7
5160 DATA 8,7,1,6,2,5,3,4
5170 DATA 4,8,5,3,6,2,7,1
5180 REM Datas pour 10 clubs
5190 DATA 1,10,2,9,3,8,4,7,5,6
5200 DATA 10,6,7,5,8,4,9,3,1,2
5210 DATA 2,10,3,1,4,9,5,8,6,7
5220 DATA 10,7,8,6,9,5,1,4,2,3
5230 DATA 3,10,4,2,5,1,6,9,7,8
5240 DATA 10,8,9,7,1,6,2,5,3,4
5250 DATA 4,10,5,3,6,2,7,1,8,9
5260 DATA 10,9,1,8,2,7,3,6,4,5
5270 DATA 5,10,6,4,7,3,8,2,9,1
5280 REM Datas pour 12 clubs
5290 DATA 1,12,2,11,3,10,4,9,5,8,6,7
5300 DATA 12,7,8,6,9,5,10,4,11,3,1,2
5310 DATA 2,12,3,1,4,11,5,10,6,9,7,8
5320 DATA 12,8,9,7,10,6,11,5,1,4,2,3
5330 DATA 3,12,4,2,5,1,6,11,7,10,8,9
5340 DATA 12,9,10,8,11,7,1,6,2,5,3,4
5350 DATA 4,12,5,3,6,2,7,1,8,11,9,10
5360 DATA 12,10,11,9,1,8,2,7,3,6,4,5
5370 DATA 5,12,6,4,7,3,8,2,9,1,10,11
5380 DATA 12,11,1,10,2,9,3,8,4,7,5,6
5390 DATA 6,12,7,5,8,4,9,3,10,2,11,1
5400 REM Datas pour 14 clubs
5410 DATA 1,14,2,13,3,12,4,11,5,10,6,9,7,8
5420 DATA 14,8,9,7,10,6,11,5,12,4,13,3,1,2
5430 DATA 2,14,3,1,4,13,5,12,6,11,7,10,8,9
5440 DATA 14,9,10,8,11,7,12,6,13,5,1,4,2,3
5450 DATA 3,14,4,2,5,1,6,13,7,12,8,11,9,10
5460 DATA 14,10,11,9,12,8,13,7,1,6,2,5,3,4
5470 DATA 4,14,5,3,6,2,7,1,8,13,9,12,10,11
5480 DATA 14,11,12,10,13,9,1,8,2,7,3,6,4,5
5490 DATA 5,14,6,4,7,3,8,2,9,1,10,13,11,12
5500 DATA 14,12,13,11,1,10,2,9,3,8,4,7,5,6
5510 DATA 6,14,7,5,8,4,9,3,10,2,11,1,12,13
```

```
5520 DATA 14,13,1,12,2,11,3,10,4,9,5,8,6,7
5530 DATA 7,14,8,6,9,5,10,4,11,3,12,2.13,1
5540 REM Datas pour 16 clubs
5550 DATA 1,16,2,15,3,14,4,13,5,12,6,11,7,10,8,9
5560 DATA 16,9,10,8,11,7,12,6,13,5,14,4,15,3,1,2
5570 DATA 2,16,3,1,4,15,5,14,6,13,7,12,8,11,9,10
5580 DATA 16,10,11,9,12,8,13,7,14,6,15,5,1,4,2,3
5590 DATA 3,16,4,2,5,1,6,15,7,14,8,13,9,12,10,11
5600 DATA 16,11,12,10,13,9,14,8,15,7,1,6,2,5,3,4
5610 DATA 4,16,5,3,6,2,7,1,8,15,9,14,10,13,11,12
5620 DATA 16,12,13,11,14,10,15,9,1,8,2,7,3,6,4,5
5630 DATA 5,16,6,4,7,3,8,2,9,1,10,15,11,14,12,13
5640 DATA 16,13,14,12,15,11,1,10,2,9,3,8,4,7,5,6
5650 DATA 6,16,7,5,8,4,9,3,10,2,11,1,12,15,13,14
5660 DATA 16,14,15,13,1,12,2,11,3,10,4,9,5,8,6,7
5670 DATA 7,16,8,6,9,5,10,4,11,3,12,2,13,1,14,15
5680 DATA 16,15,1,14,2,13,3,12,4,11,5,10,6,9,7,8
5690 DATA 8,16,9,7,10,6,11,5,12,4,13,3,14,2,15,1
5700 REM Datas pour 18 clubs
5710 DATA 1,18,2,17,3,16,4,15,5,14,6,13,7,12,8,11,9,10
5720 DATA 18,10,11,9,12,8,13,7,14,6,15,5,16,4,17,3,1,2
5730 DATA 2,18,3,1,4,17,5,16,6,15,7,14,8,13,9,12,10,11
5740 DATA 18,11,12,10,13,9,14,8,15,7,16,6,17,5,1,4,2,3
5750 DATA 3,18,4,2,5,1,6,17,7,16,8,15,9,14,10,13,11,12
5760 DATA 18,12,13,11,14,10,15,9,16,8,17,7,1,6,2,5,3,4
5770 DATA 4,18,5,3,6,2,7,1,8,17,9,16,10,15,11,14,12,13
5780 DATA 18,13,14,12,15,11,16,10,17,9,1,8,2,7,3,6,4,5
5790 DATA 5,18,6,4,7,3,8,2,9,1,10,17,11,16,12,15,13,14
5800 DATA 18,14,15,13,16,12,17,11,1,10,2,9,3,8,4,7,5,6
5810 DATA 6,18,7,5,8,4,9,3,10,2,11,1,12,17,13,16,14,15
5820 DATA 18,15,16,14,17,13,1,12,2,11,3,10,4,9,5,8,6,7
5830 DATA 7,18,8,6,9,5,10,4,11,3,12,2,13,1,14,17,15,16
5840 DATA 18,16,17,15,1,14,2,13,3,12,4,11,5,10,6,9,7,8
5850 DATA 8,18,9,7,10,6,11,5,12,4,13,3,14,2,15,1,16,17
5860 DATA 18,17,1,16,2,15,3,14,4,13,5,12,6,11,7,10,8,9
5870 DATA 9,18,10,8,11,7,12,6,13,5,14,4,15,3,16,2,17,1
5880 REM Datas pour 20 clubs
5890 DATA 1,20,2,19,3,18,4,17,5,16,6,15,7,14,8,13,9,12,10,11
5900 DATA 20,11,12,10,13,9,14,8,15,7,16,6,17,5,18,4,19,3,1,2
5910 DATA 2,20,3,1,4,19,5,18,6,17,7,16,8,15,9,14,10,13,11,12
```

```
5920 DATA 20,12,13,11,14,10,15,9,16,8,17,7,18,6,19,5,1,4,2,3
5930 DATA 3,20,4,2,5,1,6,19,7,18,8,17,9,16,10,15,11,14,12,13
5940 DATA 20,13,14,12,15,11,16,10,17,9,18,8,19,7,1,6,2,5,3,4
5950 DATA 4,20,5,3,6,2,7,1,8,19,9,18,10,17,11,16,12,15,13,14
5960 DATA 20,14,15,13,16,12,17,11,18,10,19,9,1,8,2,7,3,6,4,5
5970 DATA 5,20,6,4,7,3,8,2,9,1,10,19,11,18,12,17,13,16,14,15
5980 DATA 20,15,16,14,17,13,18,12,19,11,1,10,2,9,3,8,4,7,5,6
5990 DATA 6,20,7,5,8,4,9,3,10,2,11,1,12,19,13,18,14,17,15,16
6000 DATA 20,16,17,15,18,14,19,13,1,12,2,11,3,10,4,9,5,8,6,7
6010 DATA 7,20,8,6,9,5,10,4,11,3,12,2,13,1,14,19,15,18,16,17
6020 DATA 20,17,18,16,19,15,1,14,2,13,3,12,4,11,5,10,6,9,7,8
6030 DATA 8,20,9,7,10,6,11,5,12,4,13,3,14,2,15,1,16,19,17,18
6040 DATA 20,18,19,17,1,16,2,15,3,14,4,13,5,12,6,11,7,10,8,9
6050 DATA 9,20,10,8,11,7,12,6,13,5,14,4,15,3,16,2,17,1,18,19
6060 DATA 20,19,1,18,2,17,3,16,4,15,5,14,6,13,7,12,8,11,9,10
6070 DATA 10,20,11,9,12,8,13,7,14,6,15,5,16,4,17,3,18,2,19,1
6080 REM Datas pour le championnat de
6081 '      France 1984/85 1ere division
6082 '      Exemple avec 20 clubs
6090 DATA Nantes,Toulon,Brest,Lille,Paris SG,Nancy,Bordeaux,
Toulouse,Bastia,Monaco,Tours,Laval,Auxerre,Strasbrg,Metz,RC
Paris,Lens,Rouen,Marseill,Sochaux
6170 REM Datas pour 16 clubs
6180 DATA A,B,C,D,E,F,G,H
6190 DATA I,J,K,L,M,N,O,P
6200 REM Datas pour 14 clubs
6210 DATA A,B,C,D,E,F,G
6220 DATA H,I,J,K,L,M,N
6230 REM Datas pour 12 clubs
6240 DATA A,B,C,D,E,F
6250 DATA G,H,I,J,K,L
6260 REM Datas pour 10 clubs
6270 DATA A,B,C,D,E
6280 DATA F,G,H,I,J
6290 REM Datas pour 8 clubs
6300 DATA A,B,C,D
6310 DATA E,F,G,H
6320 REM Datas pour 6 clubs
6330 DATA A,B,C
6340 DATA D,E,F
```

```
6350 REM Datas pour 4 clubs
6360 DATA A,B
6370 DATA C,D
7000 MODE 1:INPUT"Nom du tableau";nom$
7010 OPENIN MID$(nom$,1,LEN(nom$)):INPUT#9,f:GOSUB 130
7020 FOR m=0 TO f:INPUT#9,a$(m),e(m),f(m),g(m),h(m),i(m):NEX
T m
7030 FOR m=0 TO f*2:FOR n=0 TO f/2:INPUT#9,b(m,n):NEXT n,m
7040 FOR m=0 TO f:FOR n=0 TO f*2-2:FOR o=0 TO 2:INPUT#9,c(m,
n,o):NEXT o,n,m
7050 FOR m=0 TO f-1:FOR n=0 TO f/2:FOR o=0 TO 2:INPUT#9,k(m,
n,o):NEXT o,n,m
7900 CLOSEIN:RETURN
8000 MODE 1:INPUT"Nom du tableau";nom$
8010 OPENOUT nom$:WRITE#9,f
8020 FOR m=0 TO f:WRITE#9,a$(m),e(m),f(m),g(m),h(m),i(m):NEX
T m
8030 FOR m=0 TO f*2:FOR n=0 TO f/2:WRITE#9,b(m,n):NEXT n,m
8040 FOR m=0 TO f:FOR n=0 TO f*2-2:FOR o=0 TO 2:WRITE#9,c(m,
n,o):NEXT o,n,m
8050 FOR m=0 TO f-1:FOR n=0 TO f/2:FOR o=0 TO 2:WRITE#9,k(m,
n,o):NEXT o,n,m
8900 CLOSEOUT:RETURN
10000 FOR n=1 TO f-1:CLS:PRINT "Journee";n:PRINT:PRINT:PRINT
:FOR m=1 TO f/2:PRINT m;TAB(5);a$(k(n,m,1));TAB(15);a$(k(n,m
,2));
```

CAMEROUN

========

Il est à craindre, comme c'est malheureusement le cas pour la plupart des ordinateurs familiaux, que l'essentiel des logiciels disponibles pour le CPC 464 ne soit dans une première période composé que de jeux. Nous avons essayé de montrer dans cette collection de programmes qu'on peut également faire des choses plus sérieuses avec un ordinateur et aussi qu'on peut essayer de comprendre un peu mieux comment il marche.

Mais nous n'avons rien contre les jeux et c'est pourquoi nous vous fournissons également dans ce recueil un jeu complet de Cameroun.

Nous avons ici utilisé les capacités du BASIC du CPC. C'est ainsi que nous avons créé quatre fenêtres différentes qui permettent un affichage très simple des entrées, directement avec les touches fléchées.

Nous espérons que vous connaissez les règles de ce jeu (chacun doit essayer avec 5 dés de remplir les 12 conditions prescrites). Si ce n'est pas le cas, consultez un livre de règles de jeux ou essayez simplement de déduire les règles en jouant plusieurs fois. Disons simplement que vous avez 12 tours et que vous devez essayer avec 3 lancers de dés chaque fois d'avoir finalement le plus de points possibles selon le barème du jeu.

Le CPC ne lance pas simplement les dés pour vous, il place en outre les dés dans un ordre croissant. Vous pouvez sélectionner avec la touche COPY et les touches curseur les dés que vous voulez

lancer à nouveau. Une fois que vous avez lancé les dés trois fois, sélectionnez avec les touches curseur et la touche ENTER la ligne du tableau dans laquelle vous voulez finalement placer votre résultat (si vous avez beaucoup de 6, dans la ligne des 6). Vous verrez d'autre part apparaître chaque fois le nombre global de points.

Voici comment s'effectue le décompte des points:

de 1 à 6: seuls les dés correspondant au chiffre choisi sont comptés

Brelan: si vous avez un brelan (trois fois le même chiffre), tous les dés sont additionnés

Carré: si vous avez un carré (quatre fois le même chiffre), tous les dés sont additionnés

Quinton: (cinq fois le même chiffre) 50 points

petite Suite: (au moins 3 dés se suivant) 30 points

grande Suite: (au moins 4 dés se suivant) 40 points

Full House: (un brelan et une paire, Ex: 33355) 25 points

Amusez-vous bien!

```
10 REM Cameroun
20 REM Programmes Basic pour le CPC464
30 REM Copyright 1984 DATA BECKER &
31 '      Rainer Lueers
40 INK 0,1:INK 1,24:INK 2,24
50 REM Preparatifs
60 MODE 1
70 INPUT "Nombre de joueurs (1 a 4) ";joueurs
80 IF joueurs<1 OR joueurs>4 THEN GOSUB 2350:GOTO 60
90 DIM jeu(joueurs,12),jeu1(joueurs,12)
100 FOR n=1 TO joueurs
110 PRINT:PRINT "Joueur";n;":";
120 INPUT joueur$(n)
130 IF LEN(joueur$(n))>3 THEN joueur$(n)=LEFT$(joueur$(n),3)
140 IF LEN(joueur$(n))=2 THEN joueur$(n)=joueur$(n)+" "
150 IF LEN(joueur$(n))=1 THEN joueur$(n)=joueur$(n)+"  "
160 IF LEN(joueur$(n))=0 THEN GOSUB 2350:RUN
170 NEXT n
180 MODE 1
190 REM Fenetre pour le jet des des
200 WINDOW #1,8,32,1,6
210 REM Fenetre pour tableau resultats
220 WINDOW #2,2,40,7,22
230 REM Fenetre pour informations
240 WINDOW #3,1,40,24,24
250 REM Fenetre pour choix par rapport
251 '      au tableau
260 WINDOW #4,1,1,9,21
270 REM Fixation des couleurs
280 PAPER #4,2:PEN #4,0:CLS #4:PAPER #3,2:PEN #3,0:CLS #4:CL
S #3:PAPER #1,2:PEN #1,0:CLS #1:PAPER #2,0:PEN #2,1:CLS #2:L
OCATE #4,1,13:PRINT #4,CHR$(143);
290 REM Construction du tableau
300 PRINT#2,"Nom du joueur ->  ";:FOR n=1 TO joueurs:PRINT#2
,joueur$(n);"  ";:NEXT n:PRINT#2:PRINT#2:PRINT#2,"1 --------
----":PRINT #2,"2 ------------":PRINT#2,"3 ------------":PRI
NT#2,"4 ------------":PRINT #2,"5 ------------":PRINT#2,"6 -
-----------"
310 PRINT#2,"Brelan -------":PRINT#2,"Carre --------":PRINT#
2,"Quinton ------":PRINT #2,"petite Suite -":PRINT #2,"grand
```

```
e Suite -":PRINT #2,"Full House ---":PRINT #2,STRING$(36,"="
):PRINT #2,"Points -------"
320 REM Lecture de la forme des des
321 '      en lignes de DATA
330 RESTORE:FOR n=1 TO 6:FOR m=1 TO 3
340 READ a$(n,m):NEXT m,n
350 REM Debut du programme principal:
351 '      Jeu en 12 tours
360 FOR ronde=1 TO 12
370 FOR ronde1=1 TO joueurs
380 CLS #3
390 LOCATE #3,8,1:PRINT #3,"C'est le tour de "joueur$(ronde1
)". Tour";ronde
400 REM Produire 5 nombres aleatoires
401 '      entre 1 et 6
410 FOR n=1 TO 5
420 RANDOMIZE TIME
430 a=INT(10*(RND(TIME)))
440 IF a<1 OR a>6 THEN GOTO 420
450 nombre(n)=a
460 REM Representation des des
461 '      correspondants
470 FOR m=2 TO 4
480 LOCATE #1,n*5-3,m
490 PRINT #1,a$(a,m-1);
500 NEXT m:NEXT n
510 REM Saut a la routine de tri des des
520 GOSUB 950
530 REM Chaque joueur a droit a
531 '      trois lancers par tour
540 FOR compteur=1 TO 2
550 REM Saut a la routine de
551 '      modification
560 GOSUB 820
570 CLS #1
580 FOR n=1 TO 5
590 REM Produire 5 nombres aleatoires
591 '      entre 1 et 6
600 IF wa(n)=1 THEN RANDOMIZE TIME:a=INT(10*(RND(TIME))):IF
a<1 OR a>6 THEN GOTO 600 ELSE nombre(n)=a ELSE a=nombre(n):G
```

```
OTO 620
610 REM Representation des des
611 '      correspondants
620 wa(n)=0
630 FOR m=2 TO 4
640 LOCATE #1,n*5-3,m
650 PRINT #1,a$(a,m-1);
660 NEXT m
670 NEXT n
680 REM Saut a la routine de tri des des
690 GOSUB 950
700 NEXT compteur:compteur=0
710 REM Que faut-il faire du resultat ?
720 GOSUB 1420
730 NEXT ronde1,ronde
740 REM Fin du jeu et question:
741 '      Une autre partie ?
750 a$=INKEY$
760 IF a$="" THEN GOTO 750
770 CLS
780 INPUT "Une autre partie ( /N) ";a$
790 a$=UPPER$(a$)
800 IF LEFT$(a$,1)="N" THEN END ELSE RUN
810 REM Routine de modification
820 LOCATE #1,3,5:PRINT #1,CHR$(244);
830 a$=INKEY$
840 IF a$="" THEN GOTO 830
850 REM Touche avec fleche 'gauche'
851 '      enfoncee
860 IF a$=CHR$(242) AND POS(#1)>4 THEN LOCATE #1,POS(#1)-1,5
:PRINT #1," ";:LOCATE #1,POS(#1)-6,5:PRINT #1,CHR$(244);:GOT
O 830
870 REM Touche avec fleche 'droite'
871 '      enfoncee
880 IF a$=CHR$(243) AND POS(#1)<24 THEN LOCATE #1,POS(#1)-1,
5:PRINT #1," ";:LOCATE #1,POS(#1)+4,5:PRINT #1,CHR$(244);:GO
TO 830
890 REM Touche COPY enfoncee
891 '      (effet: inversion affichage)
900 IF a$<>CHR$(224) THEN GOTO 920 ELSE IF wa((POS(#1)+1)/5)
```

```
=1 THEN wa((POS(#1)/+1)/5)=0:PEN #1,2:LOCATE #1,POS(#1)-1,1:
PRINT #1," ";:PEN #1,0:LOCATE #1,POS(#1),5 ELSE wa((POS(#1)+
1)/5)=1:LOCATE #1,POS(#1)-1,1:PRINT #1,CHR$(245);:LOCATE #1,
POS(#1),5
910 REM Touche <ENTER> enfoncee
920 IF a$=CHR$(13) THEN RETURN
930 GOTO 830
940 REM Routines de tri des des
950 FOR nn=1 TO 5
960 z2(nn)=7
970 NEXT nn
980 FOR nn=1 TO 5
990 FOR mm=1 TO 5
1000 IF nombre(mm)<=z2(nn) THEN z2(nn)=nombre(mm):z1(nn)=wa(
mm):marque=mm
1010 NEXT mm
1020 nombre(marque)=7
1030 NEXT nn
1040 FOR nn=1 TO 5
1050 nombre(nn)=z2(nn)
1060 wa(nn)=z1(nn)
1070 NEXT nn
1080 CLS #1
1090 FOR nn=1 TO 5
1100 FOR mm=2 TO 4
1110 LOCATE #1,nn*5-3,mm:PRINT #1,a$(nombre(nn),mm-1);
1120 NEXT mm
1130 NEXT nn
1140 LOCATE #1,1,6:PRINT #1,STRING$(8,CHR$(143));" Lancer";c
ompteur+1;STRING$(7,CHR$(143));
1150 RETURN
1160 REM Forme des differentes faces
1161 '      de de
1170 REM Face de de 1
1180 DATA "   "
1190 DATA " O "
1200 DATA "   "
1210 REM Face de de 2
1220 DATA "  O"
1230 DATA "   "
```

```
1240 DATA "O  "
1250 REM Face de de 3
1260 DATA "  O"
1270 DATA " O "
1280 DATA "O  "
1290 REM Face de de 4
1300 DATA "O O"
1310 DATA "   "
1320 DATA "O O"
1330 REM Face de de 5
1340 DATA "O O"
1350 DATA " O "
1360 DATA "O O"
1370 REM Face de de 6
1380 DATA "OOO"
1390 DATA "   "
1400 DATA "OOO"
1410 '      valeurs du tableau
1420 CLS #4
1430 LOCATE #4,1,13
1440 PRINT #4,CHR$(143);
1450 LOCATE #4,1,1
1460 PRINT #4,CHR$(243);
1470 a$=INKEY$
1480 IF a$="" THEN GOTO 1470
1490 REM Touche avec fleche 'bas'
1491 '      enfoncee
1500 IF a$=CHR$(241) AND VPOS(#4)<13 THEN LOCATE #4,1,VPOS(#
4)-1 ELSE GOTO 1550
1510 PRINT #4," ";
1520 LOCATE #4,1,VPOS(#4)
1530 PRINT #4,CHR$(243);
1540 REM Touche avec fleche 'haut'
1541 '      enfoncee
1550 IF a$=CHR$(240) AND VPOS(#4)>2 THEN LOCATE #4,1,VPOS(#4
)-1 ELSE GOTO 1600
1560 PRINT #4," ";
1570 LOCATE #4,1,VPOS(#4)-2
1580 PRINT #4,CHR$(243);
1590 REM Touche <ENTER> enfoncee.
```

```
1591 '      Si valeur pas encore
1592 '      enregistree, elle doit etre
1593 '      rangee et le nombre de points
1594 '      est calcule avec un
1595 '      sous-programme
1600 IF a$=CHR$(13) THEN IF jeu1(ronde1,VPOS(#4)-1)<>1 THEN
jeu1(ronde1,VPOS(#4)-1)=1:GOSUB 1820 ELSE GOTO 1470 ELSE GOT
O 1470
1610 jeu(ronde1,ronde)=resultat
1620 CLS #3
1630 LOCATE #3,9,1
1640 PRINT #3,"Joueur ";joueur$(ronde1);":";resultat;"points
";
1650 REM Inscription du resultat dans
1651 '      le tableau
1660 LOCATE #2,14+ronde1*5,VPOS(#4)+1
1670 PRINT #2,resultat;
1680 LOCATE #2,14+ronde1*5,16
1690 FOR n=1 TO 12
1700 resultatfin(ronde1)=resultatfin(ronde1)+jeu(ronde1,n)
1710 NEXT n
1720 PRINT #2,resultatfin(ronde1);
1730 resultatfin(ronde1)=0
1740 a$=INKEY$
1750 IF a$="" THEN GOTO 1740
1760 CLS #1
1770 CLS #4
1780 LOCATE #4,1,13
1790 PRINT #4,CHR$(143);
1800 CLS #3
1810 RETURN
1820 resultat=0
1830 REM Sous-programmes de calcul du
1831 '      nombre de points obtenus
1840 ON VPOS(#4)-1 GOSUB 1870,1920,1970,2020,2070,2120,2170,
2200,2230,2260,2300,2330
1850 RETURN
1860 REM Compter les 1
1861 '      (Addition de tous les 1)
1870 FOR n=1 TO 5
```

```
1880 IF nombre(n)=1 THEN resultat=resultat+1
1890 NEXT n
1900 RETURN
1910 REM Compter les 2
1911 '      (Addition de tous les 2)
1920 FOR n=1 TO 5
1930 IF nombre(n)=2 THEN resultat=resultat+2
1940 NEXT n
1950 RETURN
1960 REM Compter les 3
1961 '      (Addition de tous les 3)
1970 FOR n=1 TO 5
1980 IF nombre(n)=3 THEN resultat=resultat+3
1990 NEXT n
2000 RETURN
2010 REM Compter les 4
2011 '      (Addition de tous les 4)
2020 FOR n=1 TO 5
2030 IF nombre(n)=4 THEN resultat=resultat+4
2040 NEXT n
2050 RETURN
2060 REM Compter les 5
2061 '      (Addition de tous les 5)
2070 FOR n=1 TO 5
2080 IF nombre(n)=5 THEN resultat=resultat+5
2090 NEXT n
2100 RETURN
2110 REM Compter les 6
2111 '      (Addition de tous les 6)
2120 FOR n=1 TO 5
2130 IF nombre(n)=6 THEN resultat=resultat+6
2140 NEXT n
2150 RETURN
2160 REM Test du brelan;
2161 '      Si brelan, points de tous les
2162 '      des
2170 IF (nombre(1)=nombre(2) AND nombre(2)=nombre(3)) OR (no
mbre(2)=nombre(3) AND nombre(3)=nombre(4)) OR (nombre(3)=nom
bre(4) AND nombre(4)=nombre(5)) THEN FOR n=1 TO 5:resultat=r
esultat+nombre(n):NEXT n
```

```
2180 RETURN
2190 REM Test du carre;
2200 IF (nombre(1)=nombre(2) AND nombre(2)=nombre(3) AND nom
bre(3)=nombre(4)) OR (nombre(2)=nombre(3) AND nombre(3)=nomb
re(4) AND nombre(4)=nombre(5)) THEN FOR n=1 TO 5:resultat=re
sultat+nombre(n):NEXT n
2210 RETURN
2220 REM Test du quinton;
2221 '      Quinton => 50 points
2230 IF nombre(1)=nombre(2) AND nombre(2)=nombre(3) AND nomb
re(3)=nombre(4) AND nombre(4)=nombre(5) THEN resultat=50
2240 RETURN
2250 REM Test de la petite suite;
2251 '      Petite suite=30 points
2260 IF nombre(1)=nombre(2)-1 AND nombre(2)=nombre(3)-1 THEN
 resultat=30
2262 IF nombre(2)=nombre(3)-1 AND nombre(3)=nombre(4)-1 THEN
 resultat=30
2264 IF nombre(3)=nombre(4)-1 AND nombre(4)=nombre(5)-1 THEN
 resultat=30
2266 IF nombre(1)=nombre(4)-1 AND nombre(2)=nombre(5)-1 THEN
 resultat=30
2270 IF nombre(2)=nombre(4)-1 AND nombre(4)=nombre(5)-1 THEN
 resultat=30
2280 RETURN
2290 REM Test de la grande suite;
2291 '      Grande suite=40 points
2300 IF nombre(1)=nombre(2)-1 AND nombre(2)=nombre(3)-1 AND
nombre(3)=nombre(4)-1 THEN resultat=40
2305 IF nombre(2)=nombre(3)-1 AND nombre(3)=nombre(4)-1 AND
nombre(4)=nombre(5)-1 THEN resultat=40
2307 IF nombre(1)=nombre(3)-1 AND nombre(3)=nombre(4)-1 AND
nombre(4)=nombre(5)-1 THEN resultat=40
2310 RETURN
2320 REM Test Full House;
2321 '      Full House = 25 Points
2330 IF (nombre(1)=nombre(2) AND nombre(3)=nombre(4) AND nom
bre(4)=nombre(5)) OR (nombre(1)=nombre(2) AND nombre(2)=nomb
re(3) AND nombre(4)=nombre(5)) THEN resultat=25
2340 RETURN
```

```
2350 a$=INKEY$:IF a$="" THEN GOTO 2350 ELSE PRINT ASC(a$):GO
TO 2350
```

Décodeur

========

Ce jeu existe en de nombreuses versions, par exemple sous le nom de Mastermind.

Nous avons préféré le nom de Décodeur parce que notre programme s'éloigne malgré tout notablement de ses modèles.

Vous pouvez tout d'abord choisir parmi combien de nombres-couleurs l'ordinateur pourra sélectionner les nombres-couleurs dont doit se composer le code. Vous disposez d'un maximum de 8 codes-couleurs.

Vous devez ensuite faire un second choix: de combien d'éléments le code doit-il être composé? Il peut y avoir plus de nombres-couleurs que d'éléments du code mais pas le contraire.

L'écran affiche après chaque entrée combien de couleurs sont justes et se trouvent effectivement dans le code mais aussi combien de couleurs justes ont été également placées à la bonne place. Les débutants en Décodeur auront certainement du mal mais c'est en forgeant qu'on devient forgeron. Vous pouvez à tout moment consulter la liste de vos entrées en appuyant sur la touche ENTER.

Nous vous souhaitons maintenant de bien vous amuser avec ce jeu.

```
10 REM Sport cerebral
20 REM Programmes Basic pour le CPC464
30 REM Copyright 1984 DATA BECKER &
31 '      Rainer Lueers
40 REM Preparatifs et dimensionnements
50 DIM marque$(100),noir(100),blanc(100)
60 PAPER 0:PEN 1:MODE 1
70 INPUT"Nombre de couleurs (1 a 8) ";f$
80 IF VAL(f$)<1 OR VAL(f$)>8 THEN GOSUB 460:GOTO 70
90 INPUT"Nombre de chiffres (1 a 8) ";f1$
100 IF VAL(f1$)<1 OR VAL(f1$)>8 THEN GOSUB 460:GOTO 90
110 IF VAL(f1$)>VAL(f$) THEN GOSUB 460:GOTO 90
120 REM Choix des nombres par la
121 '      fonction de hasard RND
130 FOR n=1 TO VAL(f1$)
140 a=INT(10*RND(TIME))
150 IF a<1 OR a>VAL(f$) THEN GOTO 140
160 REM Tous les nombres choisis
161 '      doivent etre differents
170 FOR m=1 TO n-1
180 a(m)=VAL(MID$(a$,m,1))
190 NEXT m
200 FOR m=1 TO n-1
210 IF a=a(m) THEN GOTO 140 ELSE NEXT m
220 a$=a$+RIGHT$(STR$(a),1)
230 NEXT n
240 FOR n=1 TO VAL(f1$)
250 a(n)=VAL(MID$(a$,n,1))
260 NEXT n
270 CLS
280 REM Routine d'entree
290 PRINT"Un nombre avec";VAL(f1$);"chiffres ";:INPUT f3$
300 REM Listage de tous les nombres qui
301 '      ont ete entres jusqu'ici
310 IF f3$="" THEN CLS:FOR z1=1 TO z:PRINT"Tentative"z1;TAB(
8);marque$(z1);"     OK:";noir(z1);"  pas OK:";blanc(z1):NEXT
 z1:PRINT:GOTO 290 ELSE IF LEN(f3$)<>VAL(f1$) THEN GOSUB 460
:GOTO 290
320 FOR n=1 TO VAL(f1$)
330 b(n)=VAL(MID$(f3$,n,1))
```

```
340 IF a(n)=b(n) THEN noir=noir+1:c(n)=-1
350 NEXT n
360 FOR m=1 TO VAL(f1$)
370 FOR n=1 TO VAL(f1$)
380 IF a(n)=b(m) AND n<>m THEN blanc=blanc+1:GOTO 410
390 NEXT n
400 REM Routine de sortie avec controle
401 '      de la justesse
410 NEXT m:PRINT"Couleurs en mauvaise position:";blanc
420 z=z+1:marque$(z)=f3$:blanc(z)=blanc:noir(z)=noir:blanc=0
430 PRINT"Couleurs en bonne position:";noir
440 IF noir=VAL(f1$) THEN PRINT:PAPER 1:PEN 0:PRINT"Parfaite
ment exact en";z;"coups";:PAPER 0:PEN 1:PRINT:PRINT:INPUT"En
core une fois (O/ ) ";f$:IF UPPER$(LEFT$(f$,1))="O" THEN RUN
 ELSE END ELSE noir=0:GOTO 290
450 GOTO 290
460 PRINT:PRINT"Entree incorrecte!":PRINT:RETURN
```

Réflexes

========

Nous vous recommandons ce programme si vous voulez essayer de surprendre votre monde ou même pour le présenter à des personnes plutôt hostiles à tout ce qui touche aux ordinateurs.

Le but de ce programme est uniquement de mesurer le temps de réaction: il faut appuyer sur une touche quelconque du clavier dès que l'ordinateur vous le demande. Suit alors une évaluation par l'ordinateur des réflexes de la personne qui vient de jouer et on vous demande alors si vous voulez à nouveau participer à une partie avec un maximum de quatre participants.

Mais l'intérêt principal de ce programme est la façon dont les utilisateurs de l'ordinateur sont traités. Les ordinateurs, comme vous le savez certainement sont parfaitement stupides, ce sont seulement les hommes qui les ont rendus intelligents. C'est ce que ce programme a pour but de démontrer aux utilisateurs non-initiés:

Si vous n'entrez pas de nom au début du programme, le CPC ou plutôt notre programme devient grincheux et prévient qu'on va casser l'ordinateur à force de faire des erreurs. Mais si les utilisateurs font ce qui leur est demandé, l'ordinateur répond 'merci!'.

Pendant le déroulement du jeu, le CPC surveille que personne n'essaye de tricher. Si quelqu'un triche, l'ordinateur lance un avertissement et une pénalité.

Non seulement le temps de réaction est affiché précisément, au centième de seconde près, mais le CPC vous dispense également son commentaire personnel sur les performances des divers joueurs ('SUPER'...'très faible').

Sur le plan programmation, un jeu d'enfant. Mais ce programme pourra peut-être vous servir "d'excuse" auprès de ceux qui vivent avec vous en leur faisant peut-être entrevoir l'intérêt de l'achat du CPC 464! Qui sait, peut-être pourrez-vous en faire des fanas de l'informatique, ce qui serait bien sûr une belle réussite. Ceci dit, voici encore une application possible de ce programme: essayez de mesurer vos temps de réaction quand vous êtes dans des humeurs ou états d'âme différents!

```
10 REM Reaction
20 REM Programmes Basic pour le CPC464
30 REM Copyright 1984 DATA BECKER &
31 '      Rainer Lueers
40 REM Preparatifs
50 INK 0,0:INK 1,24:PEN 1:PAPER 0:MODE 1
60 INPUT"Combien de participants (1 a 4)";f$
70 IF VAL(f$)<1 OR VAL(f$)>4 THEN GOTO 60
80 FOR n=1 TO VAL(f$)
90 PRINT:PRINT"Votre nom, s'il vous plait, joueur";n:INPUT f
1$(n)
100 REM Reaction si entree incorrecte
110 IF f1$(n)="" AND marque=0 THEN PRINT:PRINT"N'ayez pas pe
ur!                              Chaque chose a son nom.
        Il en va de meme pour les hommes et les betes, si j'e
n crois mes circuits!":marque=1:GOTO 90
120 IF f1$(n)="" AND marque=1 THEN PRINT:PRINT"Je suis AMSTR
AD, mais vous pouvez        m'appeler par mon prenom: CPC, a
        condition que vous me donniez votre nom!":marque=2:GO
TO 90
140 IF f1$(n)="" AND marque=2 THEN PRINT:PRINT"Faites attent
ion: encore une entree      incorrecte et l'ordinateur sera h
.s!":marque=3:GOTO 90
150 IF f1$(n)="" AND marque=3 THEN FOR n=1 TO 10000:PRINT CH
R$(PEEK(n));:NEXT
160 marque=0
170 NEXT n
180 MODE 0:LOCATE 8,12:PRINT"Merci!"
190 FOR n=1 TO 2000:NEXT n
200 MODE 1:FOR n=1 TO VAL(f$)
210 PRINT f1$(n);"!!!!":PRINT:PRINT:PRINT"C'est a vous!":PRI
NT"Frappez une touche du clavier des que   vous verrez un ch
angement de couleur."
220 REM Calcul de l'intervale
230 FOR m=1 TO 2000:NEXT m:a=1000*RND(1):FOR m=1 TO a:a$=INK
EY$:IF a$="" THEN NEXT m:GOSUB 280 ELSE MODE 0:PRINT"Tricheu
r!!!! Il vous a fallu 5 secondes":f(n)=5:FOR m=1 TO 2000:NEX
T m:MODE 1
240 REM Sortir resultat
250 NEXT n:CLS:FOR n=1 TO VAL(f$):PRINT f1$(n);" a mis";f(n)
```

```
;"secondes!":PRINT:NEXT n:PRINT:PRINT:INPUT"Encore une parti
e ( /N) ";z$:IF LEFT$(UPPER$(z$),1)="n" THEN END
260 PRINT:INPUT"Avec les memes joueurs (O/ ) ";z$:IF LEFT$(U
PPER$(z$),1)="O" THEN GOTO 200 ELSE RUN
270 REM Sous-programmes de controle
271 '      du delai
280 MODE 0:PAPER 1:CLS:b=TIME
290 a$=INKEY$:IF a$="" THEN GOTO 290
300 c=TIME:PAPER 0:MODE 1:f(n)=(c-b)/250:PRINT f1$(n):PRINT"
Vous avez mis";f(n);"secondes!"
310 PRINT
320 REM Evaluation
330 IF f(n)<0.005 THEN PRINT"S U P E R":GOTO 370
340 IF f(n)<0.1 THEN PRINT"Tres bien":GOTO 370
350 IF f(n)<0.5 THEN PRINT"faible":GOTO 370
360 PRINT"Tres tres faible!":GOTO 370
370 FOR m=1 TO 2000:NEXT m:CLS:RETURN
```

Conversion numérique

===================

Le CPC n'affiche pas les nombres uniquement selon le système décimal qui nous est si familier puisqu'il dispose également d'instructions spéciales permettant de convertir les nombres dans d'autres systèmes décimaux: &H placé avant un nombre indique qu'il s'agit d'un nombre hexadécimal, &X qu'il s'agit d'un nombre binaire.

Entrez par exemple PRINT &HFF et vous obtiendrez la conversion en décimal de ce nombre hexadécimal: 255.

Si vous entrez maintenant PRINT &X111, vous obtiendrez la conversion en décimal de ce nombre binaire: 7.

Les difficultés commencent quand vous essayez de convertir directement un nombre hexadécimal en un nombre binaire. Il est même tout à fait impossible de travailler avec un système numérique existant autre que les systèmes hexadécimal, décimal ou binaire.

Nous avons donc réalisé ce programme pour vous permettre de vous familiariser avec les systèmes numériques. Vous pouvez convertir tout nombre que vous entrez dans chacun des trois systèmes en faisant précéder le nombre entré de la lettre H pour hexadécimal, B pour binaire, D pour décimal.

Vous pouvez en outre en appuyant sur la touche ENTER, faire convertir le nombre décimal que vous venez d'entrer en n'importe

quel système numérique (base) entre 2 et 9. Cette conversion se produit d'un façon fort intéressante puisque l'ordinateur calcule devant vous étape par étape le résultat de la conversion.

Comment celà se déroule-t-il? Prenons un exemple: comptons dans le système numérique de base 5. Appuyez sur la touche ENTER lorsqu'on vous demande "Déc., Hex. ou Bin.'. Répondez 5 à la question suivante et entrez le nombre 230. Ce nombre converti en base 5 sera '1410'.

Voyons maintenant comment l'ordinateur est arrivé à ce résultat: le système numérique de base 5 comporte les chiffres 0, 1, 2, 3 et 4. Le CPC divise le nombre 230 que vous avez entré par 5. Résultat 46, reste 0. Il divise ensuite 46 par 5: résultat 9, reste 1. 9 est à nouveau divisé par 5 car c'est encore un nombre supérieur au plus grand chiffre de ce système numérique. Le résultat est 1 avec un reste de 4. Il suffit maintenant de récapituler les valeurs des restes:

1ère division: 0 2ème division: 1 3ème division: 4

On pourrait effectuer une quatrième division: 1/5 = 0, reste 1. 1 est donc bien le quatrième reste.

En plaçant les restes de droite à gauche, nous obtenons bien le nombre 1410 comme équivalent en base 5 du nombre décimal (base 10) 230. En voici d'ailleurs la preuve:

1) 0*5^0=0 2) 1*5^1=5 3) 4*5^2=100 4) 1*5^3=125

5) 0 + 5 + 100 + 125 = 230 !

Pour être complet nous devons vous préciser qu'il est possible avec le BASIC du CPC d'effectuer, sans notre programme, des conversions directes du système décimal au système hexadécimal ou au système binaire: il faut utiliser pour celà les instructions PRINT HEX$(et PRINT BIN$(. Par contre les autres opérations de conversion numérique ne sont possibles qu'avec notre programme.

```
10 REM Convertisseur numerique
20 REM Programmes Basic pour le CPC464
30 REM Copyright 1984 DATA BECKER &
31 '      Rainer Lueers
40 INK 0,1:INK 1,24:INK 2,1,24:effet=2:normal=1
50 ON ERROR GOTO 410
60 MODE 1
70 REM Choix du systeme numerique
80 INPUT "Entree (Dec.,Hex ou Bin.) ";a$:IF VAL(a$)>65535 TH
EN GOTO 80 ELSE IF a$="" THEN GOTO 160 ELSE b$=LEFT$(a$,1):b
$=UPPER$(b$):IF b$="H" THEN GOSUB 310 ELSE IF b$="B" THEN GO
SUB 360 ELSE IF b$="D" THEN a$=RIGHT$(a$,LEN(a$)-1)
90 REM Calcul des nombres decimaux,
91 '      binaires et hexadecimaux pour
92 '      la sortie sur ecran
100 PRINT:PRINT "decimal        binaire        hexadecimal"
110 PRINT "--------------------------------------"
120 PRINT a$;TAB(9);" ";
130 b$=BIN$(VAL(a$),16):PRINT LEFT$(b$,8);" ";RIGHT$(b$,8);"
  ";
140 PRINT TAB(32);HEX$(VAL(a$),4)
150 PRINT:PRINT:GOTO 80
160 MODE 1
170 REM Choix d'un systeme numerique
171 '      de la base 2 a la base 9
180 INPUT "Quel systeme numerique (2 a 9) ";a
190 IF a=0 THEN RUN ELSE IF a<2 OR a>9 THEN GOSUB 440:GOTO 1
60
200 CLS:PRINT "Conversion DEC. - base";a:PRINT "============
============":PRINT
210 REM Calcul des nombres dans le
211 '      systeme numerique choisi (2 a 9)
220 INPUT "Votre entree (DEC.) ";b:IF b>32767 THEN GOSUB 440
:GOTO 160 ELSE PRINT
230 c=bça
240 PRINT "Nombre:";c;
250 d=b MOD a
260 PRINT TAB(15);"MOD:";d
270 a$=RIGHT$(STR$(d),1)+a$
280 IF c<a THEN a$=STR$(c)+a$:PRINT:PRINT a$:a$="":PRINT:PRI
```

```
NT:PRINT:GOTO 180
290 b=c:GOTO 230
300 REM Controler et convertir le
301 '       nombre hexadecimal entre
310 IF LEN(a$)>5 THEN RETURN
320 b$="&h "+MID$(a$,2,LEN(a$)-1)
330 IF VAL(b$)<0 THEN a$=STR$(VAL(b$)+65536) ELSE a$=STR$(VA
L(b$))
340 RETURN
350 REM Controler et convertir le
351 '       nombre binaire entre
360 IF LEN(a$)>17 THEN RETURN
370 b$="&x "+MID$(a$,2,LEN(a$)-1)
380 IF VAL(b$)<0 THEN a$=STR$(VAL(b$)+65536) ELSE a$=STR$(VA
L(b$))
390 RETURN
400 REM Routine de traitement des
401 '       erreurs
410 RESUME 420
420 RUN
430 END
440 PEN effet:PRINT:PRINT TAB(12) "Entree incorrecte!"
450 PEN normal:GOSUB 460:RETURN
460 PRINT:PRINT TAB(7) "<Appuyez sur une touche S.V.P.>"
470 f$=INKEY$:IF f$="" THEN GOTO 470
480 RETURN
```

Désassembleur

Les programmes Mémoire 1 à 5 ainsi que le programme de références des variables nous ont déjà permis de pénétrer un peu dans les "secrets internes" du CPC. Par secrets internes nous voulons dire: comment celà se passe-t-il quand un être inhumain tel que l'ordinateur arrive à penser. N'oubliez jamais à cet égard qu'un ordinateur ne peut jamais être plus intelligent qu'il n'a été programmé.

Les programmes Mémoire 1 à 5 ne nous ont cependant pas permis de rentrer suffisamment à l'intérieur de notre CPC. Nous n'avons en effet pu qu'examiner comment nos programmes BASIC sont stockés ligne par ligne, instruction par instruction dans la mémoire de l'ordinateur. Mais nous n'avons pas réussi à savoir comment l'ordinateur traite ces instructions. Nous avons notamment indiqué à cette occasion que le CPC, comme tout autre ordinateur ramène tout à l'alternative la plus simple, éteint ou allumé, oui ou non, 0 ou 1. C'est en effet seulement à partir d'une suite de 0 et de 1 que l'ordinateur arrive à reconnaître un caractère ou une instruction.

Notre BASIC est un langage de programmation évolué: il est en effet structuré en un nombre important de mots intelligibles (surtout quand on comprend l'anglais). Ces instructions nous permettent d'avoir beaucoup plus facilement une idée de ce que peut faire l'ordinateur que si nous ne pouvions entrer que des 0 et des 1 dans notre ordinateur.

Le code assembleur auquel s'attaque ce programme constitue une étape intermédiaire entre le BASIC et le langage binaire de la machine. Nous disposons ici d'u nombre encore plus grand d'instructions (le processeur Z80A du CPC possède plus de 600 instructions) mais ces instructions sont très loin d'être aussi puissantes que les instructions BASIC. Ces instructions occupent 1, 2 ou 3 octets en mémoire. Le programme 'Désassembleur' connait tous les codes assembleur du Z80A (ces codes sont aussi appelés mnémoniques) et il recherche dans la zone que nous voulons quels codes il rencontre. Il recherche pour celà s'il rencontre des suites de valeurs déterminées (de même que notre programme Mémoire recherchait le mot 'Examen'). S'il rencontre ces suites de nombres, il écrit le code assembleur correspondant sur l'écran.

Comme nous ne pensons pas que vous allez vous mettre à écrire du jour au lendemain un grand nombre de programmes en langage-machine, nous avons intégré une fonction pédagogique dans notre 'Désassembleur': la ligne 3530 contient en effet une instruction DATA suivie de nombres hexadécimaux qui constituent un petit programme en langage-machine. Ce programme calcule le résultat de deux nombres et le place dans un adresse déterminée. Vous pouvez regarder comment ces nombres se transforment en un programme assembleur en répondant, une fois le programme lancé, à toutes les questions en appuyant simplement sur la touche ENTER.

Vous pouvez aussi écrire d'autres nombres (uniquement en format hexadécimal!) dans cette ligne DATA, par exemple si vous trouvez dans une revue d'informatique un long programme en langage-machine pour le processeur Z80A. Vous pouvez d'ailleurs écrire plusieurs lignes de DATA si nécessaire. Amusez-vous bien!

```
10 REM Desassembleur
20 REM Programmes Basic pour le CPC464
30 REM Copyright 1984 DATA BECKER &
31 '      Rainer Lueers
40 CLEAR
50 REM Liberer la zone memoire a partir
51 '      de 42001 pour des routines
52 '      en langage-machine
60 MEMORY 42000:GOSUB 3450
70 DIM i$(255),bl(255):n=0:m=0:sy=1
80 GOTO 1280
90 MODE 2:c=0:PRINT TAB(20)"* * * Desassembleur Z-80 * * *"
100 REM Determiner les souhaits de
101 '      l'utilisateur
110 PRINT:INPUT "Titre ";f$
120 IF LEN(f$)>80 THEN GOTO 110
130 PRINT
140 REM Adresse de depart de la routine
141 '      en langage-machine stockee
142 '      en lignes DATA
150 INPUT "Adresse de depart (en dec.) ";a:IF a<0 OR a>65535
 THEN GOTO 130 ELSE IF a=0 THEN a=42001
160 INPUT "Adresse de fin (en dec.)    ";b
170 IF b<e OR b>65535 OR b<0 THEN GOTO 160 ELSE IF b=0 THEN
b=compteur1
180 PRINT:INPUT "Y a-t-il des zones de donnees (O/ ) ";v$
190 v$=LEFT$(UPPER$(v$),1):IF v$="O" THEN GOSUB 2140
200 PRINT:INPUT "Imprimante ou Ecran (I/E) ";b$
210 REM Determination de la variable
211 '      drubi qui fixe le type de sortie
212 '      avec PRINT:
213 '      8-> imprimante, 0-> ecran
220 b$=LEFT$(UPPER$(b$),1):IF b$="I" THEN drubi=8 ELSE drubi
=0
230 PRINT #drubi
240 REM Titre apparaissant en milieu
241 '      de ligne
250 PRINT #drubi,TAB((80-LEN(f$))/2-5)f$
260 PRINT #drubi
270 REM Debut du desassembleur
```

```
271 '      proprement dit. Lecture
272 '      de la zone de la memoire
273 '      indiquee et interpretation des
274 '      donnees en mnemoniques
275 '      d'instructions en langage-mach.
280 q=a
290 h1=INT(a/4096):h2=INT((a-h1*4096)/256)
300 l1=INT((a-h1*4096-h2*256)/16)
310 l2=a-h1*4096-h2*256-l1*16
320 IF m<>0 THEN GOTO 340
330 GOTO 390
340 IF a>=a(n) AND a<=b(n) THEN GOTO 360
350 GOTO 390
360 d=PEEK(a):GOSUB 3380:fh$="":fl$="":gh$="":gl$="":t$=i$(d
):i$(d)="DATA"
370 IF a=b(n) THEN n=n+1:m=m-1:GOTO 400
380 GOTO 400
390 d=PEEK(a):GOSUB 650
400 v=d:GOSUB 2740
410 dh$=h$:dl$=l$
420 IF h1>=10 THEN GOTO 560
430 h1$=STR$(h1):h1$=MID$(h1$,2,1)
440 IF h2>=10 THEN GOTO 570
450 h2$=STR$(h2):h2$=MID$(h2$,2,1)
460 IF l1>=10 THEN GOTO 580
470 l1$=STR$(l1):l1$=MID$(l1$,2,1)
480 IF l2>=10 THEN GOTO 590
490 l2$=STR$(l2):l2$=MID$(l2$,2,1)
500 REM Sortie de la zone de la memoire
501 '      desassemblee
510 qt=qt+1:IF qt>105 THEN qt=0:FOR qi=1 TO 9:PRINT #drubi:N
EXT qi:GOSUB 3400:sy=sy+1:PRINT #drubi,TAB(30) "Page";sy:PRI
NT #drubi:GOTO 510 ELSE PRINT #drubi,q;LEN(dh$+dl$+eh$+el$+f
h$+fl$+gh$+gl$)/2;"Octets";TAB(18);
520 PRINT #drubi,USING "    !!!!";" ",h1$,h2$,l1$,l2$;:PRINT
#drubi,"               ";:PRINT #drubi,USING "!!!!!!!!";" ",dh$,
dl$,eh$,el$,fh$,fl$,gh$,gl$;:PRINT #drubi,TAB(50);i$(d)
530 a=a+1:c=c+1:IF a>=b+1 THEN GOTO 600
540 IF gh$="T" THEN i$(d)=t$
550 GOTO 280
```

```
560 x=h1-10+65:h1$=CHR$(x):GOTO 440
570 x=h2-10+65:h2$=CHR$(x):GOTO 460
580 x=l1-10+65:l1$=CHR$(x):GOTO 480
590 x=l2-10+65:l2$=CHR$(x):GOTO 510
600 PRINT:PRINT "* *  Sortie terminee  * *"
610 REM Demander s'il faut recommencer
620 PRINT:INPUT "Encore une fois (O/ ) ";b$:b$=LEFT$(UPPER$(
b$),1)
630 IF b$="O" THEN RUN
640 END
650 IF d<64 OR d>127 THEN GOTO 710
660 IF d=118 THEN RETURN
670 dh=INT(d/16):dl=d-dh*16
680 g=dl AND 7:f=((d AND 56)/8)
690 i$(d)="LD         "+j$(f)+","
700 i$(d)=i$(d)+j$(g):GOSUB 3380:RETURN
710 IF d<128 OR d>191 THEN GOTO 830
720 dh=INT(d/16):dl=d-(dh*16)
730 g=dl AND 7:f=((d AND 120)/8)
740 IF f=0 THEN i$(d)="ADD        A,"
750 IF f=1 THEN i$(d)="ADC        A,"
760 IF f=2 THEN i$(d)="SUB        "
770 IF f=3 THEN i$(d)="SBC        A,"
780 IF f=4 THEN i$(d)="AND        "
790 IF f=5 THEN i$(d)="XOR        "
800 IF f=6 THEN i$(d)="OR         "
810 IF f=7 THEN i$(d)="CP         "
820 i$(d)=i$(d)+j$(g):GOSUB 3380:RETURN
830 IF bl(d)=1 THEN GOTO 880
840 IF bl(d)=2 THEN GOTO 890
850 IF bl(d)=3 THEN GOTO 900
860 IF bl(d)=4 THEN GOTO 1770
870 IF d=221 OR d=253 THEN GOTO 2220
880 GOSUB 3380:RETURN
890 a=a+1:z=PEEK(a):GOTO 910
900 a=a+1:z=PEEK(a):a=a+1:z1=PEEK(a)
910 v=z:GOSUB 2740:eh$=h$:el$=l$
920 IF bl(d)=3 THEN GOTO 940
930 GOSUB 3390:GOTO 950
940 v=z1:GOSUB 2740:fh$=h$:fl$=l$
```

```
950 p1$=fh$+fl$+eh$+el$:p2$=eh$+el$:gh$="":gl$=""
960 f=d AND 7:g=d AND 56:g=g/8
970 IF f=6 AND (d AND 192)=0 THEN GOTO 1230
980 IF f=2 AND (d AND 192)=192 THEN GOTO 1260
990 IF f=4 THEN GOTO 1270
1000 IF f=0 AND (g<>2) THEN GOTO 1240
1010 i$(33)="LD         HL,"+p1$
1020 i$(34)="LD         ("+p1$+"),HL"
1030 i$(50)="LD         ("+p1$+"),A"
1040 i$(205)="CALL       "+p1$
1050 i$(195)="JP         "+p1$
1060 i$(58)="LD         A,("+p1$+")"
1070 i$(254)="CP         "+p2$
1080 i$(42)="LD         HL,("+p1$+")"
1090 i$(49)="LD         SP,"+p1$
1100 i$(17)="LD         DE,"+p1$
1110 i$(16)="DJNZ       "+p2$
1120 i$(1)="LD         BC,"+p1$
1130 i$(198)="ADD        A,"+p2$
1140 i$(206)="ADC        A,"+p2$
1150 i$(211)="OUT        "+p2$+",A"
1160 i$(214)="SUB        "+p2$
1170 i$(219)="IN         A,"+p2$
1180 i$(222)="SBC        A,"+p2$
1190 i$(230)="AND        "+p2$
1200 i$(238)="XOR        "+p2$
1210 i$(246)="OR         "+p2$
1220 RETURN
1230 i$(d)="LD         "+j$(g)+","+p2$:RETURN
1240 IF g=3 THEN i$(d)="JR         "+p2$:RETURN
1250 g=g-4:i$(d)="JR         "+s$(g)+","+p2$:RETURN
1260 i$(d)="JP         "+s$(g)+","+p1$:RETURN
1270 i$(d)="CALL       "+s$(g)+","+p1$:RETURN
1280 FOR s=0 TO 63:bl(s)=1:NEXT s
1290 FOR s=192 TO 255:bl(s)=1:NEXT s
1300 bl(118)=1:bl(6)=2:bl(14)=2:bl(16)=2:bl(22)=2:bl(24)=2:b
l(30)=2:bl(32)=2:bl(38)=2:bl(40)=2:bl(46)=2:bl(48)=2:bl(54)=
2:bl(56)=2
1310 bl(62)=2:bl(198)=2:bl(206)=2:bl(211)=2:bl(214)=2:bl(219
)=2:bl(222)=2:bl(230)=2:bl(238)=2:bl(246)=2:bl(254)=2
```

```
1320 bl(1)=3:bl(17)=3:bl(33)=3:bl(34)=3:bl(42)=3:bl(49)=3:bl
(50)=3:bl(58)=3:bl(194)=3:bl(195)=3:bl(196)=3:bl(202)=3:bl(2
04)=3:bl(205)=3:bl(210)=3:bl(212)=3:bl(218)=3:bl(220)=3
1330 bl(226)=3:bl(228)=3:bl(234)=3:bl(236)=3:bl(242)=3:bl(24
4)=3:bl(250)=3:bl(252)=3:bl(203)=3:bl(237)=3:bl(221)=0:bl(25
3)=3
1340 j$(0)="B":j$(1)="C":j$(2)="D":j$(3)="E":j$(4)="H":j$(5)
="L"
1350 j$(6)="(HL)":j$(7)="A"
1360 s$(0)="NZ":s$(1)="Z":s$(2)="NC":s$(3)="C":s$(4)="PO":s$
(5)="PE":s$(6)="P":s$(7)="M"
1370 p$(0)="SBC        HL,":p$(1)="ADC        HL,"
1380 p$(2)="SBC        HL,":p$(3)="ADC        HL,"
1390 p$(4)="SBC        HL,":p$(5)="ADC        HL,"
1400 p$(7)="ADC        HL,":n$(0)="BC":n$(1)="BC":n$(2)="DE"
1410 n$(3)="DE":n$(4)="HL":n$(5)="HL":n$(7)="SP"
1420 l$(0)="N":l$(1)="I":m$(0)="0":m$(2)="1":m$(3)="2"
1430 o$(0)="I,A":o$(1)="R,A":o$(2)="A,I":o$(3)="A,R"
1440 q$(0)="LD":q$(1)="CP":q$(2)="IN":q$(3)="OUT"
1450 i$(0)="NOP":i$(2)="LD         (BC),A":i$(3)="INC        B
C"
1460 i$(7)="RLC        A":i$(8)="EX         AF,AF'"
1470 i$(9)="ADD        HL,BC":i$(10)="LD         A,(BC)"
1480 i$(11)="DEC        BC":i$(15)="RRC        A"
1490 i$(18)="LD         (DE),A":i$(19)="INC        DE"
1500 i$(23)="RLA":i$(25)="ADD        HL,DE":i$(26)="LD
 A,(DE)"
1510 i$(27)="DEC        DE":i$(31)="RRA":i$(35)="INC        HL
"
1520 i$(39)="DAA":i$(41)="ADD        HL,HL":i$(43)="DEC
 HL"
1530 i$(47)="CPL":i$(51)="INC        SP":i$(55)="SCF"
1540 i$(57)="ADD        HL,SP":i$(59)="DEC        SP":i$(63)="
CCF"
1550 i$(43)="DEC        HL"
1560 i$(192)="RET        NZ":i$(197)="PUSH       BC"
1570 i$(199)="RST        0":i$(200)="RET        Z":i$(207)="RS
T        8"
1580 i$(208)="RET        NC":i$(209)="POP        DE"
1590 i$(213)="PUSH       DE":i$(215)="RST        10H"
```

```
1600 i$(216)="RET       C":i$(223)="RST       18H"
1610 i$(224)="RET       PO":i$(227)="EX        (SP),HL"
1620 i$(231)="RST       20H":i$(232)="RET       PE"
1630 i$(233)="JP        (HL)":i$(235)="EX        DE,HL"
1640 i$(239)="RST       28H":i$(240)="RET       P"
1650 i$(201)="RET"
1660 i$(241)="POP       AF":i$(243)="DI":i$(245)="PUSH
AF"
1670 i$(247)="RST       30H":i$(248)="RET       M":i$(251)="
EI"
1680 i$(249)="LD        SP,HL":i$(255)="RST       38H"
1690 i$(197)="PUSH      BC"
1700 i$(193)="POP       BC":i$(229)="PUSH      HL"
1710 i$(225)="POP       HL"
1720 i$(217)="EXX"
1730 i$(118)="HALT"
1740 l=0:FOR d=5 TO 45 STEP 8:i$(d)="DEC       "+j$(l):l=l+1
:NEXT d
1750 l=0:FOR d=4 TO 44 STEP 8:i$(d)="INC       "+j$(l):l=l+1
:NEXT d
1760 i$(60)="INC       A":i$(61)="DEC       A":GOTO 90
1770 IF d=203 THEN GOTO 1790
1780 IF d=237 THEN GOTO 2950
1790 a=a+1:z=PEEK(a)
1800 eh=INT(z/16):el=z-eh*16:g=el AND 7
1810 f=INT((el AND 8)/8):f=f+(eh*2):GOSUB 1830
1820 GOTO 1980
1830 IF f=0 THEN i$(d)="RLC       ":RETURN
1840 IF f=1 THEN i$(d)="RRC       ":RETURN
1850 IF f=2 THEN i$(d)="RL        ":RETURN
1860 IF f=3 THEN i$(d)="RR        ":RETURN
1870 IF f=4 THEN i$(d)="SLA       ":RETURN
1880 IF f=5 THEN i$(d)="SRA       ":RETURN
1890 IF f=7 THEN i$(d)="SRL       ":RETURN
1900 IF f>=8 AND f<=15 THEN GOTO 1940
1910 IF f>=16 AND f<=23 THEN GOTO 1960
1920 f=f AND 7:f$=CHR$(f+48)
1930 i$(d)="SET       "+f$+",":RETURN
1940 f=f AND 7:f$=CHR$(f+48)
1950 i$(d)="BIT       "+f$+",":RETURN
```

```
1960 f=f AND 7:f$=CHR$(f+48)
1970 i$(d)="RES        "+f$+",":RETURN
1980 IF eh>=10 THEN GOTO 2030
1990 eh$=STR$(eh):eh$=MID$(eh$,2,1)
2000 IF el>=10 THEN GOTO 2040
2010 el$=STR$(el):el$=MID$(el$,2,1)
2020 GOSUB 3390:i$(d)=i$(d)+j$(g):RETURN
2030 x=eh+55:eh$=CHR$(x):GOTO 2000
2040 x=el+55:el$=CHR$(x):GOTO 2020
2050 IF d AND 7=5 THEN GOTO 2080
2060 IF d AND 7=4 THEN GOTO 2080
2070 GOTO 880
2080 l=d AND 56
2090 g=l/8
2100 IF d AND 7=5 THEN i$(d)="DEC        "+j$(g):GOTO 2120
2110 i$(d)="INC        "+j$(g)
2120 GOSUB 3380:RETURN
2130 REM Sous-programme pour exclure
2131 '      certaines zones de la memoire
2132 '      du desassemblage
2140 n=1
2150 INPUT "Adresse de depart=";a(n)
2160 INPUT "Adresse de fin=";b(n)
2170 IF a(n)<a OR b(n)>b THEN GOTO 2150
2180 INPUT "Encore une zone de donnees (O/ ) ";v$
2190 v$=LEFT$(UPPER$(v$),1)
2200 IF v$="O" THEN n=n+1:GOTO 2150
2210 m=n:n=1:RETURN
2220 IF d=221 THEN v$="IX":GOTO 2240
2230 v$="IY"
2240 a=a+1:z=PEEK(a)
2250 IF z=203 THEN GOTO 2660
2260 IF z>=70 AND z<=190 THEN GOTO 2810
2270 IF z=33 OR z=34 OR z=42 THEN GOTO 2430
2280 IF z=52 OR z=53 THEN GOTO 2530
2290 IF z=54 THEN GOTO 2590
2300 v=z:GOSUB 2740:eh$=h$:el$=l$:GOSUB 3390
2310 IF z=9 THEN i$(d)="ADD        "+v$+",BC"
2320 IF z=25 THEN i$(d)="ADD        "+v$+",DE"
2330 IF z=35 THEN i$(d)="INC        "+v$
```

```
2340 IF z=41 THEN i$(d)="ADD        "+v$+","+v$
2350 IF z=43 THEN i$(d)="DEC        "+v$
2360 IF z=57 THEN i$(d)="ADD        "+v$+",SP"
2370 IF z=225 THEN i$(d)="POP        "+v$
2380 IF z=227 THEN i$(d)="EX         (SP),"+v$
2390 IF z=229 THEN i$(d)="PUSH       "+v$
2400 IF z=233 THEN i$(d)="JP         ("+v$+")"
2410 IF z=249 THEN i$(d)="LD         SP,"+v$
2420 RETURN
2430 v=z:GOSUB 2740
2440 eh$=h$:el$=l$
2450 a=a+1:z1=PEEK(a):v=z1:GOSUB 2740
2460 fh$=h$:fl$=l$:a=a+1
2470 z1=PEEK(a):v=z1:GOSUB 2740
2480 gh$=h$:gl$=l$
2490 IF z=33 THEN i$(d)="LD         "+v$+","+gh$+gl$+fh$+fl$
2500 IF z=34 THEN i$(d)="LD         ("+gh$+gl$+fh$+fl$+"),"+v
$
2510 IF z=42 THEN i$(d)="LD         "+v$+",("+gh$+gl$+fh$+fl$
+")"
2520 RETURN
2530 v=z:GOSUB 2740:eh$=h$:el$=l$
2540 a=a+1:z1=PEEK(a):v=z1:GOSUB 2740
2550 fh$=h$:fl$=l$:gh$="":gl$=""
2560 IF z=52 THEN i$(d)="INC         ("+v$+"+"+fh$+fl$+")"
2570 IF z=53 THEN i$(d)="DEC         ("+v$+"+"+fh$+fl$+")"
2580 RETURN
2590 v=z:GOSUB 2740:eh$=h$:el$=l$
2600 a=a+1:z1=PEEK(a):v=z1:GOSUB 2740
2610 fh$=h$:fl$=l$:a=a+1
2620 z1=PEEK(a):v=z1:GOSUB 2740
2630 gh$=h$:gl$=l$
2640 i$(d)="LD         ("+v$+"+"+fh$+fl$+"),"+gh$+gl$
2650 RETURN
2660 a=a+1:eh$=CHR$(67):el$=CHR$(66)
2670 v=PEEK(a):GOSUB 2740
2680 fh$=h$:fl$=l$
2690 a=a+1:z2=PEEK(a):o=z2 AND 248:o=o/8
2700 q$="("+v$+"+"+fh$+fl$+")"
2710 f=o:GOSUB 1830
```

```
2720 i$(d)=i$(d)+q$
2730 v=z2:GOSUB 2740:gh$=h$:gl$=l$:RETURN
2740 h=INT(v/16):l=v-(h*16)
2750 IF h>=10 GOTO 2790
2760 h$=STR$(h):h$=MID$(h$,2,1)
2770 IF l>=10 THEN GOTO 2800
2780 l$=STR$(l):l$=MID$(l$,2,1):RETURN
2790 x1=h+55:h$=CHR$(x1):GOTO 2770
2800 x1=l+55:l$=CHR$(x1):RETURN
2810 v=z:GOSUB 2740:eh$=h$:el$=l$:gh$="":gl$=""
2820 a=a+1:z1=PEEK(a)
2830 v=z1:GOSUB 2740:fh$=h$:fl$=l$
2840 IF z=126 THEN i$(d)="LD         A,("+v$+"+"+fh$+fl$+")":
RETURN
2850 p=z AND 240
2860 IF p=112 THEN GOTO 2900
2870 IF p>=128 THEN GOTO 2920
2880 p=z AND 56:p=p/8:GOSUB 3290
2890 i$(d)="LD          "+g$+",("+v$+"+"+fh$+fl$+")":RETURN
2900 p=z AND 7:GOSUB 3290
2910 i$(d)="LD          ("+v$+"+"+fh$+fl$+"),"+g$:RETURN
2920 p=z AND 56:p=p/8:GOSUB 3290
2930 i$(d)=i$(d)+v$+"+"+fh$+fl$+")"
2940 RETURN
2950 a=a+1:z=PEEK(a):v=z:GOSUB 2740:eh$=h$:el$=l$
2960 IF z=67 OR z=75 OR z=83 OR z=91 OR z=115 OR z=123 THEN
GOTO 3200
2970 GOSUB 3390:f=z AND 248:g=z AND 7
2980 IF f=160 THEN GOTO 3160
2990 IF f=168 THEN GOTO 3170
3000 IF f=176 THEN GOTO 3180
3010 IF f=184 THEN GOTO 3190
3020 f=z AND 56:f=f/8:g=z AND 7
3030 IF f=6 THEN i$(d)="SBC        HL,SP":RETURN
3040 IF g=0 THEN i$(d)="IN         "+j$(f)+",(C)":RETURN
3050 IF g=1 THEN i$(d)="OUT        (C),"+j$(f):RETURN
3060 IF g=2 THEN i$(d)=p$(f)+n$(f):RETURN
3070 IF g=4 THEN i$(d)="NEG":RETURN
3080 IF g=5 THEN i$(d)="RET"+l$(f):RETURN
3090 IF g=6 THEN i$(d)="IM "+m$(f):RETURN
```

```
3100 IF g<>7 THEN i$(d)="* * *":RETURN
3110 IF f<=3 THEN GOTO 3150
3120 IF f=4 THEN i$(d)="RRD":RETURN
3130 IF f=5 THEN i$(d)="RLD":RETURN
3140 GOTO 3100
3150 i$(d)="LD          "+o$(f):RETURN
3160 i$(d)=q$(g)+"I":RETURN
3170 i$(d)=q$(g)+"D":RETURN
3180 i$(d)=q$(g)+"IR":RETURN
3190 i$(d)=q$(g)+"DR":RETURN
3200 a=a+1:z1=PEEK(a):v=z1:GOSUB 2740:fh$=h$:fl$=l$
3210 a=a+1:z1=PEEK(a):v=z1:GOSUB 2740:gh$=h$:gl$=l$
3220 gg$=gh$+gl$+fh$+fl$
3230 IF z=67 THEN i$(d)="LD          ("+gg$+"),BC":RETURN
3240 IF z=75 THEN i$(d)="LD          BC,("+gg$+")":RETURN
3250 IF z=83 THEN i$(d)="LD          ("+gg$+"),DE":RETURN
3260 IF z=91 THEN i$(d)="LD          DE,("+gg$+")":RETURN
3270 IF z=115 THEN i$(d)="LD          ("+gg$+"),SP":RETURN
3280 IF z=123 THEN i$(d)="LD          SP,("+gg$+")":RETURN
3290 IF p=0 THEN g$="B":i$(d)="ADD          A,("
3300 IF p=1 THEN g$="C":i$(d)="ADC        A,("
3310 IF p=2 THEN g$="D":i$(d)="SUB       ("
3320 IF p=3 THEN g$="E":i$(d)="SBC       A,("
3330 IF p=4 THEN g$="H":i$(d)="AND       ("
3340 IF p=5 THEN g$="L":i$(d)="XOR       ("
3350 IF p=6 THEN g$="***":i$(d)="OR        ("
3360 IF p=7 THEN g$="A":i$(d)="CP         ("
3370 RETURN
3380 eh$="":el$="":fh$="":fl$="":gh$="":gl$="":RETURN
3390 fh$="":fl$="":gh$="":gl$="":RETURN
3400 PRINT #drubi:RETURN
3410 PRINT #drubi:qt=0:sy=1
3420 PRINT #drubi:RETURN
3430 REM Fin de la bibliotheque de
3431 '      sous-programmes pour creer
3432 '      le code assembleur
3440 REM Verifier le nombre d'octets
3441 '      (entree en hexadecimal) a
3442 '      desassembler en lignes DATA
3450 ON ERROR GOTO 3540:FOR compteur1=42001 TO 65535:READ a$
```

```
:NEXT compteur1
3460 REM Lecture proprement dite des
3461 '      donnees en lignes DATA et
3462 '      ecriture de ces donnees
3463 '      a partir de 42001
3470 RESTORE:FOR m=42001 TO compteur1-1
3480 READ a$
3490 a=VAL("&h"+a$)
3500 POKE m,a
3510 NEXT m:POKE m,&C9
3520 RETURN
3530 DATA 3e,04,06,07,80,32,00,7d
3540 RESUME 3470
```

Les TOKENS du BASIC du CPC (voir Mémoire 1 à 5) :

128 AFTER	129 AUTO	130 BORDER	131 CALL
132 CAT	133 CHAIN	134 CLEAR	135 CLG
136 CLOSEIN	137 CLOSEOUT	138 CLS	139 CONT
140 DATA	141 DEF	142 DEFINT	143 DEFREAL
144 DEFSTR	145 DEG	146 DELETE	147 DIM
148 DRAW	149 DRAWR	150 EDIT	151 ELSE
152 END	153 ENT	154 ENV	155 ERASE
156 ERROR	157 EVERY	158 FOR	159 GOSUB
160 GOTO	161 IF	162 INK	163 INPUT
164 KEY	165 LET	166 LINE	167 LIST
168 LOAD	169 LOCATE	170 MEMORY	171 MERGE
172 MID$	173 MODE	174 MOVE	175 MOVER
176 NEXT	177 NEW	178 ON	179 ON BREAK
180 ON ERROR GOTO 0		181 ON SQ	182 OPENIN
183 OPENOUT	184 ORIGIN	185 OUT	186 PAPER
187 PEN	188 PLOT	189 PLOTR	190 POKE
191 PRINT	192 '		
193 RAD	194 RANDOMIZE	195 READ	196 RELEASE
197 REM	198 RENUM	199 RESTORE	200 RESUME
201 RETURN	202 RUN	203 SAVE	204 SOUND
205 SPEED	206 STOP	207 SYMBOL	208 TAG
209 TAGOFF	210 TROFF	211 TRON	212 WAIT
213 WEND	214 WHILE	215 WIDTH	216 WINDOW
217 WRITE	218 ZONE	219 DI	220 EI
221 à 226 : -			
227 ERL	228 FN	229 SPC	230 STEP
231 SWAP	232 ET 233 : -		234 TAB
235 THEN	236 TO	237 USING	238 >
239 =	240 >=	241 <	242 <>
243 <=	244 +	245 -	246 *
247 /	248 ^	249 \	250 AND
251 MOD	252 OR	253 XOR	254 NOT

Avec le "préfixe" 255, on a les codes suivants :

0 ABS	1 ASC	2 ATN	3 CHR$
4 CINT	5 COS	6 CREAL	7 EXP
8 FIX	9 FRE	10 INKEY	11 INP
12 INT	13 JOY	14 LEN	15 LOG
16 LOG10	17 LOWER$	18 PEEK	19 REMAIN
20 SGN	21 SIN	22 SPACE$	23 SQ
24 SQR	25 STR$	26 TAN	27 UNT
28 UPPER$	29 VAL		
30 à 60 : -			
64 EOF	65 ERR	66 HIMEM	67 INKEY$
68 PI	69 RND	70 TIME	71 XPOS
72 YPOS	73 à 112 : -		
113 BIN$	114 DEC$	115 HEX$	116 INSTR
117 LEFT$	118 MAX	119 MIN	120 POS
121 RIGHT$	122 ROUND	123 STRING$	124 TEST
125 TESTR	126		127 VPOS

POUR LES CPC 664 ET 6128 :

TOKEN	INSTRUCTION
DD	FILL
DF	MASK
E0	FRAME
E1	CURSOR
DE	GRAPHICS

CE LIVRE VOUS PASSIONNE,

MAIS VOUS N'AVEZ PAS LE TEMPS

DE TAPER LES PROGRAMMES

ACHETEZ LA DISQUETTE

DES PROGRAMMES DU LIVRE

PROGRAMMES BASIC POUR LES CPC !

NOM : ______________________________

PRENOM : ______________________________

ADRESSE : ______________________________

____

CODE POSTAL : __________ **VILLE :** ____________________

Je désire recevoir la disquette des programmes du livre PROGRAMMES BASIC POUR LES CPC.

Je joins à ce coupon un [] chèque de 120 francs.
[] CCP de 120 francs.

Date : **Signature :**

MICRO APPLICATION MICRO APPLICATION

MICRO APPLICATION MICRO APPLICATION

TEXTOMAT AMSTRAD
CPC 464 & 664

Traitement de texte de qualité professionnelle pour tous.
Tabulation, recherche, remplacement, insertion, manipulation de paragraphes, calcul... Accents à l'écran et imprimante. Module permettant de gérer tout type d'imprimante. Ecrit en LANGAGE MACHINE. Liaison avec DATAMAT pour mailing et lettres types personnalisées... TEXTOMAT est la solution traitement de texte sur CPC. Documentation complète.

Réf. : AM305
Prix : 450 FF





DATAMAT AMSTRAD
CPC 464 & 664

La gestion de fichier la plus complète fonctionnant pour les 464 et 664. Entièrement en LANGAGE MACHINE. Fonctions de calcul, de tri, de recherche multicritères, impressions paramètrables, liaison avec TEXTOMAT pour mailing... Documentation française de 60 pages.

Réf. : AM304
Prix : 450 FF

D.A.M.S. POUR AMSTRAD
CPC 464 & 664

D.A.M.S. est un logiciel intégrant un assembleur, un moniteur et un désassembleur symbolique pour développer et mettre au point facilement des programmes en langage machine sur les micro ordinateurs AMSTRAD. Les trois modules sont co-résidents en mémoire ce qui assure une grande souplesse d'utilisation. Vous pouvez notamment utiliser un éditeur plein écran, un assembleur immédiat, un désassembleur symbolique, une trace et beaucoup d'autres fonctions très puissantes. D.A.M.S. est entièrement relogeable et est bien évidemment écrit en langage machine.

Réf. : AM208
Prix : sur cassette : 295 FF TTC pour CPC 464
Réf. : AM308
Prix : sur disquette : 395 FF TTC pour CPC 664 & CPC 464




MICRO INFO
T23
XB2
ATARI
COMMODORE
COMMODORE
AMSTRAD
80 pages de trucs et astuces,
programmes, dossiers
pour Amstrad CPC,
Commodore
Atari ST ...
20 f

Achevé d'imprimer en février 1986
sur les presses de l'imprimerie Laballery et Cie
58500 Clamecy
Dépôt légal : février 1986
Numéro d'imprimeur : 602027

IMPRIMÉ EN FRANCE

[redacted]

Ce livre a pour but d'alimenter votre CPC en super-programmes écrits en basic. Différents types de programmes (jeux, utilitaires, applications...) vous permettront de vous familiariser avec l'utilisation de l'informatique mais aussi avec la programmation. Enfin vous pourrez même faire vos premiers pas en langage machine.

Principaux sujets abordés :

— A la découverte de la mémoire

— Facilitez-vous le travail avec des EDITEURS
- Editeur graphique
- Editeur sonore
- Editeur de texte

— Programmez plus facilement en BASIC
- Jeux de caractères (français, mathématiques, informatiques)
- Messages d'erreurs complets
- Références des variables

— Des programmes utiles
- Calendrier
- Gestion d'une collection (disques par exemple)
- Suivi d'un championnat sportif (foot, basket...)

— Des jeux
- Jeu de dés
- Décodeur
- Réflexes

— Premiers pas en langage machine
- Conversion numérique
- Désassembleur

— Les tokens du BASIC

Nouvelle édition :
Les programmes de ce livre ont été entièrement revus et adaptés pour fonctionner sur les ordinateurs **AMSTRAD CPC 464, 664 et 6128.**

ISBN : 2-86899-007-X
PRIX : 129 FF

Réf. : ML 119


AMSTRAD
PROGRAMMES BASIC
POUR LES CPC
2
MA


AMSTRAD
CPC
OCR
MÉMOIRE ÉCRITE
MEMORY ENGRAVED
MEMORIA ESCRITA
ACME
https://acpc.me/